



Universidad
Carlos III de Madrid

Departamento de Informática

PROYECTO FIN DE CARRERA

Desarrollo de la aplicación Servidor del juego Cuatrola para Android

Autor: Jorge Juan Muñoz Bautista

Tutor: Lorena González Manzano

Leganés, octubre de 2015

Título: Desarrollo de la aplicación Servidor del juego Cuatrola para Android
Autor: Jorge Juan Muñoz Bautista
Director: Lorena González Manzano

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día __ de _____
de 20__ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de
Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Agradecimientos

Sois muchas las personas a las que me gustaría dar mis más sinceros agradecimientos por el apoyo recibido durante todo el camino recorrido hasta llegar aquí. Aunque no pueda nombraros a todo por no hacer este capítulo interminable, quiero que sepáis que me acuerdo de cada uno de vosotros.

Quiero agradecer a mi familia, y en especial a mis padres y hermana, todo el apoyo incondicional mostrado, porque si no fuera por ellos no habría llegado hasta aquí. A mis tíos y abuelos por animarme y ayudarme cuando lo necesitaba.

A mi novia Elena por todo su amor, generosidad, apoyo y en especial por compartir su vida conmigo.

A todos mis profesores y compañeros de clase por su paciencia y ayuda.

A mi grupo de amigos Dani, Sebas, Juanjo, Juan Carlos, Luis, Andrea, Belén, Laura y Pablo por todo lo vivido y disfrutado en estos años.

A mi compañero y buen amigo, Álvaro, por su ayuda, comprensión, paciencia y tenacidad.

A mi tutora Lorena por ayudarme, animarme y por el esfuerzo realizado para que hoy pueda escribir este documento.

Y por último, a la persona más especial de mi vida, mi hijo Marcos, por hacerme sacar fuerzas cuando no las tengo, por alegrarme con tan solo verle sonreír y sobre todo por hacerme feliz.

Gracias a todos los que me habéis acompañado durante este largo viaje por ayudarme, comprenderme, aguantarme y sobre todo por hacer de esta experiencia algo inolvidable.

Índice general

1. INTRODUCCIÓN Y OBJETIVOS	11
1.1 Introducción	12
1.2 Motivación	15
1.3 Objetivos	16
1.4 Organización del presente documento	17
2. ESTADO DEL ARTE	19
2.1 Panorámica de aplicaciones existentes.....	20
2.2 Aplicación actual de la Cuatrola	21
3. ANÁLISIS	23
3.1 Perspectiva general del sistema.....	24
3.2 Reglas de la Cuatrola	25
3.3 Arquitectura del sistema.....	27
3.4 Estudio tecnológico.....	29
3.4.1 Tecnologías impuestas.....	29
3.4.2 Tecnologías aplicables al componente Base de Datos	29
3.4.3 Tecnologías aplicables a los componentes Gestión Usuarios, Gestión Mesas y Gestión Juego.....	30
3.4.4 Tecnologías asociadas al servidor.....	31
3.5 Selección de tecnologías	32
3.6 Arquitectura definitiva de alto nivel	33
3.7 Casos de uso	35
3.7.1 Diagramas de casos de uso.....	35
3.7.2 Definición textual de los casos de uso	36
3.8 Requisitos de software	40
3.8.1 Requisitos funcionales	40
3.8.2 Requisitos no funcionales	41
3.9 Diseño del plan de pruebas de aceptación.....	42
4. DISEÑO DETALLADO.....	45
4.1 Diseño de Software	46
4.1.1 Componente Base de datos	47

ÍNDICE general

4.1.2 Componente Gestión de Usuarios	48
4.1.3 Componente Gestión de Mesas.....	50
4.1.4 Componente Gestión Juego	50
4.1.5 Componente Gestión Sincronización.....	53
4.2 Diagramas de secuencia	54
4.2.1 Insertar usuario (CU-01).....	54
4.2.2 Validar Usuario (CU-02)	55
4.2.3 Insertar Mesa (CU-03)	55
4.2.4 Asociar Usuario Mesa (CU-04).....	56
4.2.5 Recuperar partida (CU-05)	57
4.2.6 Jugar partida (CU-06).....	58
4.2.7 Enviar Estadísticas (CU-07).....	60
5. IMPLEMENTACIÓN DEL SOFTWARE	61
5.1 Diseño de Software	62
5.1.1 Sincronización con GCM.....	62
5.1.2 Uso de JSON.....	63
5.1.3 Uso de DNS dinámico.....	63
5.2 Resultados de las pruebas de aceptación	65
6. CONCLUSIONES Y LÍNEAS FUTURAS.....	66
6.1 Conclusiones sobre el proyecto	67
6.1.1 Resultados obtenidos	67
6.1.2 Dificultades del proyecto.....	68
6.1.3 Conclusiones personales	69
6.2 Líneas futuras	70
6.2.1 Parametrización de los máxima puntuación para finalizar una partida.....	70
6.2.2 Creación de mesas públicas	70
6.2.3 Creación de chat para la partida	70
6.2.4 Aumentar las estadísticas proporcionadas.....	71
6.2.5 Introducción de un ranking de puntuaciones.	71
6.2.6 Introducción de diferentes juegos de cartas.	71
6.2.7 Creación de una aplicación web para jugar a Cuatrola.....	71
REFERENCIAS.....	73
ACRÓNIMOS	77
ANEXO 1: GESTIÓN DEL PROYECTO	79
ANEXO2: MANUAL DE INSTALACIÓN Y CONFIGURACIÓN DEL SERVIDOR.....	98
ANEXO 3: PLANTILLAS	108

Índice de figuras

Figura 1 – Número de aplicaciones disponibles según la tienda de apps [2].....	13
Figura 2 - Captura de pantalla de los comentarios de la Cuatrola del Google Play Store	22
Figura 3 – Arquitectura preliminar detallada del sistema	28
Figura 4 – Arquitectura definitiva de alto nivel	33
Figura 5 – Diagrama de casos de uso	35
Figura 6 - Diagrama de componentes definitivo	46
Figura 7 – Modelo relacional de la base de datos	47
Figura 8 – Modelo de clases del componente Gestión de Usuarios.....	49
Figura 9 – Modelo de clases del componente Gestión de Mesas.....	50
Figura 10 – Modelo de clases del componente Gestión del Juego	52
Figura 11 – Modelo de clases del componente Gestión Sincronización.....	53
Figura 12 – Diagrama de secuencia de insertar usuario (CU-01)	54
Figura 13 – Diagrama de secuencia de validar usuario (CU-02)	55
Figura 14 – Diagrama de secuencia de insertar mesa (CU-03).....	56
Figura 15 – Diagrama de secuencia de asociar usuario mesa (CU-04).....	57
Figura 16 – Diagrama de secuencia de recuperar mesa (CU-05).....	58
Figura 17 – Diagrama de secuencia de jugar partida (CU-06).....	59
Figura 18 – Diagrama de secuencia de enviar estadísticas (CU-07).....	60
Figura 19 - Planificación inicial del proyecto	82
Figura 20 - Desarrollo real del proyecto	85
Figura 21 – Aviso de XAMPP por posibles problemas con el antivirus	99
Figura 22 – Ventana selección de componentes XAMPP	100
Figura 23 – Panel de control de XAMPP	101
Figura 24 – Archivo Configuración Apache	102
Figura 25 – Consola web de administración de MySQL	102
Figura 26 – Importación de fichero en la consola de MySQL	103
Figura 27 – Pantalla registro No-Ip.....	104
Figura 28 – Ventana configuración No - Ip	105
Figura 29 – Asociar host en No-Ip	106
Figura 30 – Cliente No-Ip arrancado y configurado	106

Índice de tablas

Tabla 1 - Definición textual del caso de uso CU-01	36
Tabla 2 - Definición textual del caso de uso CU-02	37
Tabla 3 - Definición textual del caso de uso CU-03	37
Tabla 4 - Definición textual del caso de uso CU-04	38
Tabla 5 - Definición textual del caso de uso CU-05	38
Tabla 6 - Definición textual del caso de uso CU-06	39
Tabla 7 - Definición textual del caso de uso CU-07	39
Tabla 8 – Requisitos funcionales	41
Tabla 9 – Requisitos no funcionales	41
Tabla 10 – Pruebas de aceptación	44
Tabla 11 - Resultados de las pruebas de aceptación	65
Tabla 12 – Planificación inicial del proyecto detallada	80
Tabla 13 – Desarrollo real del proyecto detallado	83
Tabla 14 – Comparativa entre planificación estimada y desarrollo real del proyecto	86
Tabla 15 – Herramientas utilizadas en el proyecto	88
Tabla 16 – Equipos utilizados en el proyecto	89
Tabla 17 - Gastos de personal	91
Tabla 18 - Gastos de equipos	92
Tabla 19 - Gastos de software	93
Tabla 20 - Gastos de consumibles.....	93
Tabla 21 - Gastos de viajes y dietas	94
Tabla 22 - Costes directos	95
Tabla 23 – Estimación de costes	95
Tabla 24 - Presupuesto para el cliente.....	96
Tabla 25 - Coste real del proyecto comparado con el presupuestado	97
Tabla 26 - Plantilla para la definición de casos de uso	108
Tabla 27 - Plantilla para la especificación de requisitos de software.....	109
Tabla 28 - Plantilla para la especificación de pruebas de aceptación	109
Tabla 29 - Plantilla para los resultados de las pruebas de aceptación.....	110

Capítulo 1

Introducción y objetivos

1.1 Introducción

A lo largo de estos últimos años hemos sido testigos de un gran avance tecnológico en el mundo de los teléfonos móviles. Sus capacidades de procesamiento y almacenamiento, así como sus funciones, han crecido enormemente. Los teléfonos móviles han pasado de servir solo para recibir y realizar llamadas a ser utilizados prácticamente para todo. Por este motivo reciben el nombre de teléfonos inteligentes o, en inglés, smartphones.

La gran utilidad y las diferentes funcionalidades que ofrecen los teléfonos móviles inteligentes han supuesto que éstos hayan tenido una gran acogida entre la población mundial. Según el informe “Internet Trends 2015 – Code Conference” publicado por la empresa de capitales de riesgo KPCB, en 2015 habrá en torno a 1.600 millones de smartphones en el mundo [1].

Además de las diferentes funcionalidades que traen estos dispositivos de fábrica, se les pueden instalar todo tipo de aplicaciones que ofrecen multitud de nuevas funcionalidades. En la actualidad existe un gran número de aplicaciones en el mercado, ya sean gratuitas o de pago, que ofrecen diferentes servicios: ya sea para la comunicación, para el trabajo, para facilitar las tareas cotidianas o incluso para el entretenimiento. En la figura que se muestra a continuación se puede ver la cantidad de aplicaciones existentes en las diferentes tiendas de aplicaciones de las plataformas para dispositivos móviles más usadas:

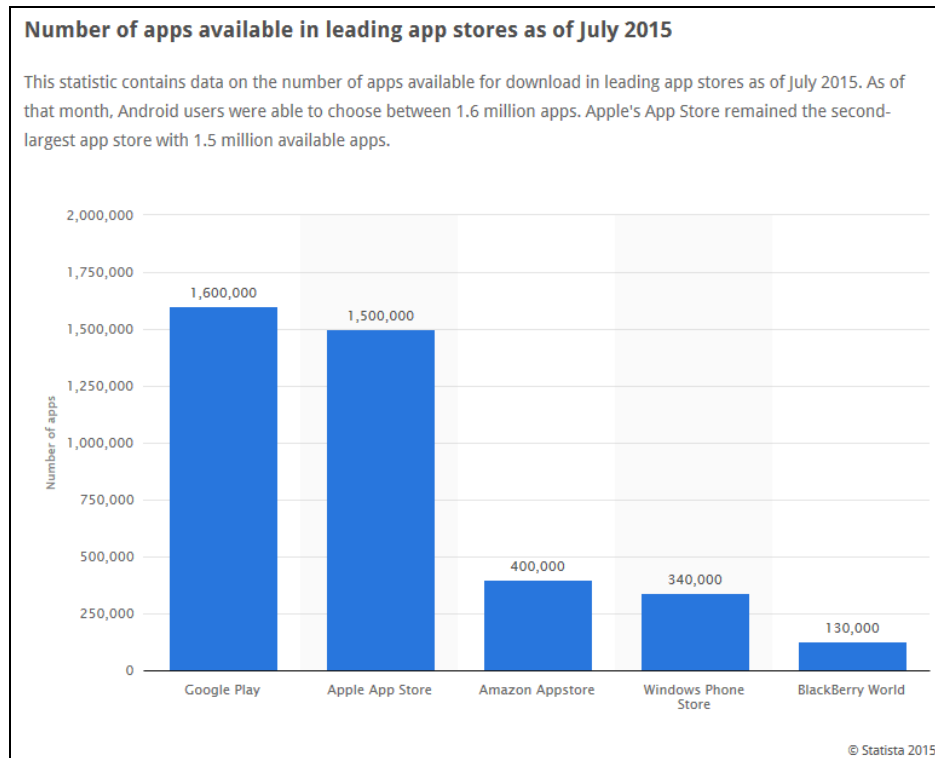


Figura 1 – Número de aplicaciones disponibles según la tienda de apps [2]

El desarrollo de aplicaciones para dispositivos móviles es un mercado en continuo crecimiento, llegando a ser uno de los más importantes en la actualidad. Es tal el crecimiento, que el número de aplicaciones disponibles para dispositivos móviles ha aumentado de manera exponencial en los últimos años, llegando a los valores vistos en la Figura 1. Según podemos ver en la figura anterior, el número de aplicaciones disponibles para descargar actualmente en la tiendas de aplicaciones de las dos plataformas móviles más importantes, Android y Apple, supera los tres millones de aplicaciones.

Entre las aplicaciones más descargadas por los usuarios, se encuentran los juegos. Según el último informe sobre juegos móviles publicado por eMarketer, empresa dedicada al estudio del mercado digital, el número de descargas de juegos para plataformas móviles en Estados Unidos registrará un crecimiento del 16,5%, lo que supondrá un total de 3.040 millones de dólares en ingresos [3]. Es por este motivo por lo que este proyecto se va a centrar en el desarrollo de un juego de entretenimiento para Smartphones y en concreto en un juego de cartas.

Existen multitud de aplicaciones para móviles basadas en juego de cartas, desde los típicos juegos de baraja española como el mus y el tute a juegos de baraja francesa o americana como el póker.

Este proyecto fin de carrera se va centrar en el desarrollo de un juego de cartas de baraja española. Tras realizar un estudio de las aplicaciones disponibles de este tipo, se ha detectado que no existe ninguna aplicación para poder jugar al juego de cartas españolas de la Cuatrola de manera online contra otros usuarios, por lo que este proyecto se encargará de cubrir esta necesidad.

1.2 Motivación

Tras realizar un estudio de las aplicaciones existentes actualmente, se ha detectado la falta del juego de la Cuatrola para poder jugar con otras personas en tiempo real a través de una conexión a internet. Actualmente existen aplicaciones para poder jugar a este juego pero sólo permiten jugar contra la máquina, lo que hace que el juego no enganche y los usuarios se cansen rápido de jugar. Cuando se participa en una partida en cualquier juego contra personas reales, y más aún si estas personas son conocidas, la motivación por ganar crece, lo que provoca que este tipo de juegos mantengan el interés del jugador durante más tiempo.

Teniendo en cuenta las características de las aplicaciones actuales que permiten jugar a la Cuatrola, la motivación principal para el desarrollo de este proyecto es conseguir que los usuarios puedan jugar contra otras personas a través de su dispositivo móvil, teniendo una conexión a internet y por tanto, jugar en tiempo real.

De esta manera, estaremos creando un juego con una característica bien diferenciada respecto a los actuales y tendrá su parte de mercado dentro de las aplicaciones existentes para dispositivos móviles.

1.3 Objetivos

Este proyecto fin de carrera se realizará de manera conjunta con otro compañero. La aplicación se divide en dos partes, la parte cliente y la parte servidor. La primera será desarrollada en el proyecto de Álvaro Fernández Morales, mientras que la segunda, la parte servidor será desarrollada en este proyecto.

El objetivo, considerando tanto la parte cliente como la parte servidor, es el análisis, diseño e implementación de una aplicación para dispositivo móvil que permita a los usuarios jugar al juego de cartas españolas de la Cuatrola en tiempo real, siempre y cuando tengan conexión a internet.

Además, se han establecido dos sub-objetivos:

- La generación de unas estadísticas personalizadas por usuario: en las que se mostrará el número de las partidas ganadas y perdidas, así como el número de solos, cuatrolas y quintolas tanto ganadas como perdidas. Esto último indicará al jugador si arriesga demasiado al apostar y jugar el solo contra la pareja contraria.
- El almacenamiento del estado de la partida cada vez que haya algún movimiento. Así, se podrá parar la partida y retomar más tarde, cuando los jugadores puedan.

1.4 Organización del presente documento

Con el objetivo de facilitar la lectura del documento en esta sección se presenta una breve descripción del contenido de cada una de las partes que forman este documento, compuesto por seis capítulos y tres anexos:

Capítulo 1, Introducción y objetivos: En este capítulo se ofrece una introducción a las aplicaciones para dispositivo móvil, junto con las motivaciones de este proyecto fin de carrera y los objetivos que se persiguen.

Capítulo 2, Estado del arte: en este capítulo se muestra un análisis de las principales aplicaciones que se han desarrollado hasta la actualidad del juego de la Cuatrola.

Capítulo 3, Análisis: En este capítulo se expone una perspectiva general del sistema junto con el estudio de la arquitectura que tendrá el sistema, definiéndose los componentes que la integrarán. El capítulo también incluye un estudio de las tecnologías a utilizar en el desarrollo del sistema junto con la especificación de los casos de uso, los requisitos de software y las pruebas de aceptación definidas para comprobar el cumplimiento de estos requisitos.

Capítulo 4, Diseño detallado: En este capítulo se detallan los componentes definidos en el capítulo de análisis mostrando sus diagramas de clases. Adicionalmente se mostrarán un conjunto de diagramas de secuencia que permitan comprender las distintas interacciones que tendrán lugar entre los distintos componentes y clases del sistema.

Capítulo 5, Implementación y pruebas: En este capítulo se presentan los detalles más destacables de la implementación del sistema junto con los resultados de las pruebas de aceptación llevadas a cabo.

Capítulo 6, Conclusiones y líneas futuras: Este último capítulo expone las conclusiones obtenidas con el desarrollo del proceso en distintos ámbitos así como alguna de las líneas de desarrollo futuras a seguir para extender la funcionalidad del sistema o mejorar su funcionamiento.

Anexo 1, Gestión del proyecto: En este anexo se detalla la planificación del proyecto junto con el seguimiento del mismo. Adicionalmente se muestra el presupuesto del proyecto junto con las desviaciones detectadas en el mismo.

Anexo 2, Manual de usuario: En este anexo se muestra el manual de usuario en el que se detalla como instalar y configurar el servidor.

Anexo 3, Plantillas: En este anexo se muestran las plantillas utilizadas para recoger distintos datos a lo largo de la memoria.

Capítulo 2

Estado del arte

2.1 Panorámica de aplicaciones existentes

En esta sección se muestra el estudio realizado de las aplicaciones, para dispositivos móviles, disponibles hoy en día para poder jugar a la Cuatrola.

Se ha realizado una búsqueda exhaustiva en las tiendas de aplicaciones más utilizadas actualmente y a continuación se expone el resultado obtenido en cada una de ellas:

Apple App Store desde la aplicación de iTunes [4] → no existe ninguna aplicación para poder jugar a la Cuatrola en iOS (el sistema operativo de Apple).

Google Play Store [5]

En cuanto a las aplicaciones existentes para Android: existen dos, que en realidad es la misma, pero una gratis con anuncios y otra de pago sin ellos.

Amazon App Store[6] → en la tienda de aplicaciones de Amazon tampoco aparece ninguna aplicación para poder jugar a la Cuatrola.

BlackberryWorld [7] → en la tienda para BlackBerry aparece la misma aplicación que en el Google Play Store para Android, en este caso, únicamente de pago.

Microsoft Windows Phone App Store [8] → para Windows Phone tampoco existe ninguna aplicación para poder jugar a este juego.

Tras ver estos resultados podemos concluir que únicamente se ha creado una aplicación para dispositivo móvil con la cual poder jugar a la Cuatrola. Esta aplicación se puede descargar desde el Google Play Store (gratis con anuncios o de pago) y desde BlackberryWorld de pago. En el siguiente apartado se detallará esta aplicación.

2.2 Aplicación actual de la Cuatrola

En este apartado explicaremos cómo es la única aplicación que existe actualmente para poder jugar a la Cuatrola desde un dispositivo móvil. Esta app es la misma que se puede descargar en el Play Store y en el BlackberryWorld como hemos visto en el apartado anterior.

A continuación se detallarán las características de esta aplicación:

En esta app al entrar solamente te ofrece dos opciones:

1. ¿Cómo jugar? → en el que se explican las reglas y como está dispuesta la mesa de juego.
2. Jugar → donde comienza al instante una partida en la que únicamente se puede jugar contra la máquina, no contra ninguna persona real. Esto hace que sea menos apetecible jugar durante bastante tiempo.

En cuanto al número de descargas en BlackBerry World no facilitan la información, pero en el Google Play Store si lo hacen, de manera aproximada. Los datos son: aplicación gratuita con anuncios: de 50.000 a 100.000 y la aplicación de pago, que cuesta 0,50€, de 100 a 500 descargas.

Aun así los comentarios de los usuarios indican, a parte de algunos errores en la programación, la importante necesidad de que este juego fuera online para poder jugar contra otras personas. La mayoría de errores han sido provocados porque la Cuatrola tiene una gran cantidad de variantes y de obligaciones que tienen que cumplir los jugadores debido a sus reglas de juego, las cuales se han explicado anteriormente.

En la siguiente página se muestra una figura con la captura de pantalla en la que se puede ver la buena puntuación de la aplicación y varios comentarios en los que los usuarios solicitan que se pueda jugar online contra otros jugadores. Precisamente esto es lo que vamos a conseguir con la realización de este proyecto.

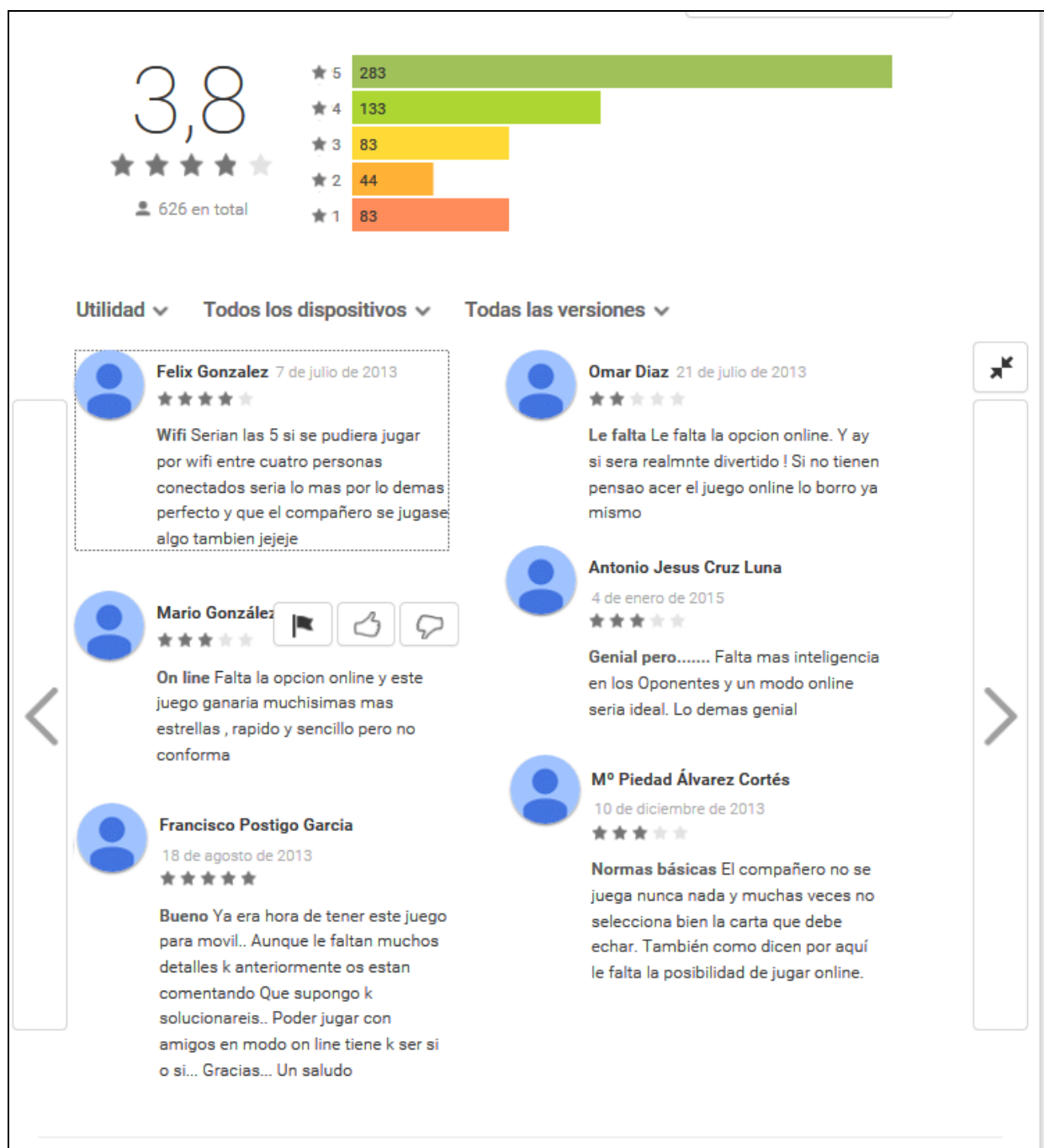


Figura 2 - Captura de pantalla de los comentarios de la Cuatrola del Google Play Store

Capítulo 3

Análisis

3.1 Perspectiva general del sistema

En este capítulo se describe el sistema completo que se implementará para cumplir con el principal objetivo propuesto en el Capítulo 1 de este documento: desarrollar una aplicación para dispositivos móviles que permita jugar al juego de naipes Cuatrola desde varios dispositivos móviles, conectados a una misma partida.

El sistema a desarrollar deberá facilitar la generación y recreación de partidas de Cuatrola permitiendo a los usuarios jugar entre sí, en tiempo real, desde sus teléfonos móviles. Como la idea es simular una partida real, se deberán cumplir y seguir cada una de las reglas que contiene dicho juego. Las reglas del juego se expondrán en el siguiente apartado (3.2 Reglas de la Cuatrola).

Para acometer este objetivo, será necesario que el sistema permita a los diferentes dispositivos móviles unirse a una misma partida y comunicarse entre sí, así como, que diferentes usuarios puedan estar jugando a la vez que se están disputando otras partidas. Por lo tanto, será necesario usar alguna tecnología y arquitectura que permita la comunicación entre los diferentes dispositivos móviles.

Como todo juego, el principal propósito de la aplicación es entretener y alcanzar a la mayor cantidad de público posible. Por lo que el acceso, generación y recreación de partidas tendrán que realizarse de un modo simple, rápido e intuitivo. También, deberá ser accesible desde un gran número de teléfonos móviles diferentes. Además de lo anterior, es importante que un juego haga que el usuario no se aburra y juegue el máximo de veces. Para alcanzar este propósito, la aplicación tendrá que almacenar la información de las partidas y jugadas ganadas y perdidas por el usuario, para que éste pueda ir mejorando y superándose.

Uno de los mayores inconvenientes de este tipo de juegos es que dependen de la disponibilidad de los usuarios. Para evitar este problema el sistema posibilitará la opción de salirse de una partida y poder continuarla en otro momento.

3.2 Reglas de la Cuatrola

La Cuatrola es un juego de baraja española, de la familia del tute, jugado por parejas que compiten entre sí. Es, también, un juego de apuesta. Como en el tute, los jugadores se colocan, cada uno enfrente de su pareja, quedando, de esta forma, intercalados con los contrarios. En este juego de las 48 cartas de la baraja española solo se utilizan 20: las sotas, los caballos, los reyes, los treses y los ases de los cuatro palos: oros, copas, espadas y bastos.

Inicio de la partida

Una partida se inicia repartiendo las cartas y mostrando la última carta que se reparte, que será el pinte. Esta carta indica cual es el palo del triunfo.

Una vez realizado esto, cada jugador estudiará sus cartas y planteará su apuesta de quintola (ganar las cinco bazas de que consta la mano jugando solo), Cuatrola (ganar, al menos, cuatro bazas de las cinco que hay en la mano jugando el solo), un solo (el apostante gana la mano sin la ayuda del compañero) o no jugarse nada. Cada jugador deberá expresar su apuesta en orden, empezando por el que está a la derecha del que repartió, la mano, y terminando con el jugador que repartió. Cada apuesta debe superar a las apuestas previas.

Tras haber decidido que se juega cada uno, únicamente se jugará la apuesta mayor, empezando a tirar la mano su primera carta.

Si un jugador se juega algo, su pareja no participa en esa mano (se le salta el turno).

Si tocaba empezar la mano a un jugador que no participa (porque su pareja se juega algo), el inicio de la baza salta al jugador de su derecha. Continúan el resto de jugadores siempre en sentido anti horario.

Desarrollo del juego

- 1.- La mano abre la baza tirando cualquier carta.
- 2.- Se continúa el juego, por parte del resto de los jugadores, tirando cartas que superen la anterior en valor y del mismo palo. En caso de no tener cartas que superen lo que hay en la mesa, se está obligado a asistir (tirar una carta del mismo palo).

3.- Si no se puede asistir, por no tener ese palo, es obligatorio fallar con una carta de triunfo. El fallo de un jugador posterior deberá superar al del anterior. Cuando no se puede cumplir con nada de lo anterior se echa cualquier carta. Si hay un fallo anterior a tu turno, ya no es obligatorio montar la carta que inició la baza, pero si asistir cuando sea posible.

4.- La baza la ganará el jugador que haya lanzado la carta de más valor si todos han asistido al palo. Si ha habido fallos con cartas de triunfo, gana la mano el jugador que haya echado el único triunfo o, si hay varias cartas de triunfo, gana la mano el jugador que tiró el triunfo de más valor.

Puntuación

* En cuatrolas y quintolas las manos se ganan en función de las bazas ganadas.

* En solos y manos sin apuesta, gana la pareja que sume más puntos según la siguiente puntuación:

- Los ases acumulados suman 11 puntos; los treses, 10 puntos; los reyes, 4 puntos; los caballos, 3 puntos y las sotas 2 puntos.
- El jugador que gana la última baza suma 10 puntos más.
- A estas puntuaciones hay que sumar los cantes que suman 20 puntos extra, o 40 puntos extra si son de triunfo (tener el caballo y el rey de un palo y ganar una baza; en ese momento se canta y se anota a la puntuación). Los cantes solo tienen sentido cuando está en juego un solo o una mano sin apuesta.

* En la cuenta final del juego: Una partida sin apuesta representa un punto.

* Un solo representa dos puntos.

* Una Cuatrola son cuatro puntos. Si el apostante pierde dos bazas, los cuatro puntos los suma la pareja adversaria.

* Una quintola son cinco puntos. Si el apostante pierde una baza, los cinco puntos los suma la pareja adversaria.

* La partida se termina y es ganada cuando una de las parejas consigue, exactamente, 20 puntos. Esto quiere decir, por ejemplo, que cuando te queda solamente un punto para llegar a los 20 no te puedes apostar nada. La única excepción que permite rebasar los 20 puntos es aquella situación en que los puntos últimos provienen de la pareja adversaria por perder una apuesta.

3.3 Arquitectura del sistema

Como hemos visto anteriormente, el sistema deberá permitir la conexión y comunicación entre varios dispositivos móviles. Es por ello, por lo que se deberá utilizar una arquitectura que soporte esta necesidad. Para dar solución a este problema, se ha decidido utilizar una arquitectura cliente-servidor. Dentro de esta arquitectura este proyecto se encargará de realizar la parte servidor.

Este es un modelo de procesamiento cooperativo entre dos o más máquinas; las que solicitan recursos o servicios, llamadas clientes, y las proveedoras, llamadas servidores. Los clientes solicitan un determinado servicio a un servidor, y éste envía mensajes con la respuesta, realizando este proceso de manera transparente al usuario.

El servidor se encarga de atender las solicitudes de los clientes. Entre sus principales funciones están: acceder y manejar los recursos de datos (normalmente bases de datos), realizar validaciones a nivel de base de datos, procesar la lógica de la aplicación y formatear los datos para que sean legibles por los clientes.

A continuación se muestra el diagrama preliminar que detalla el sistema completo:

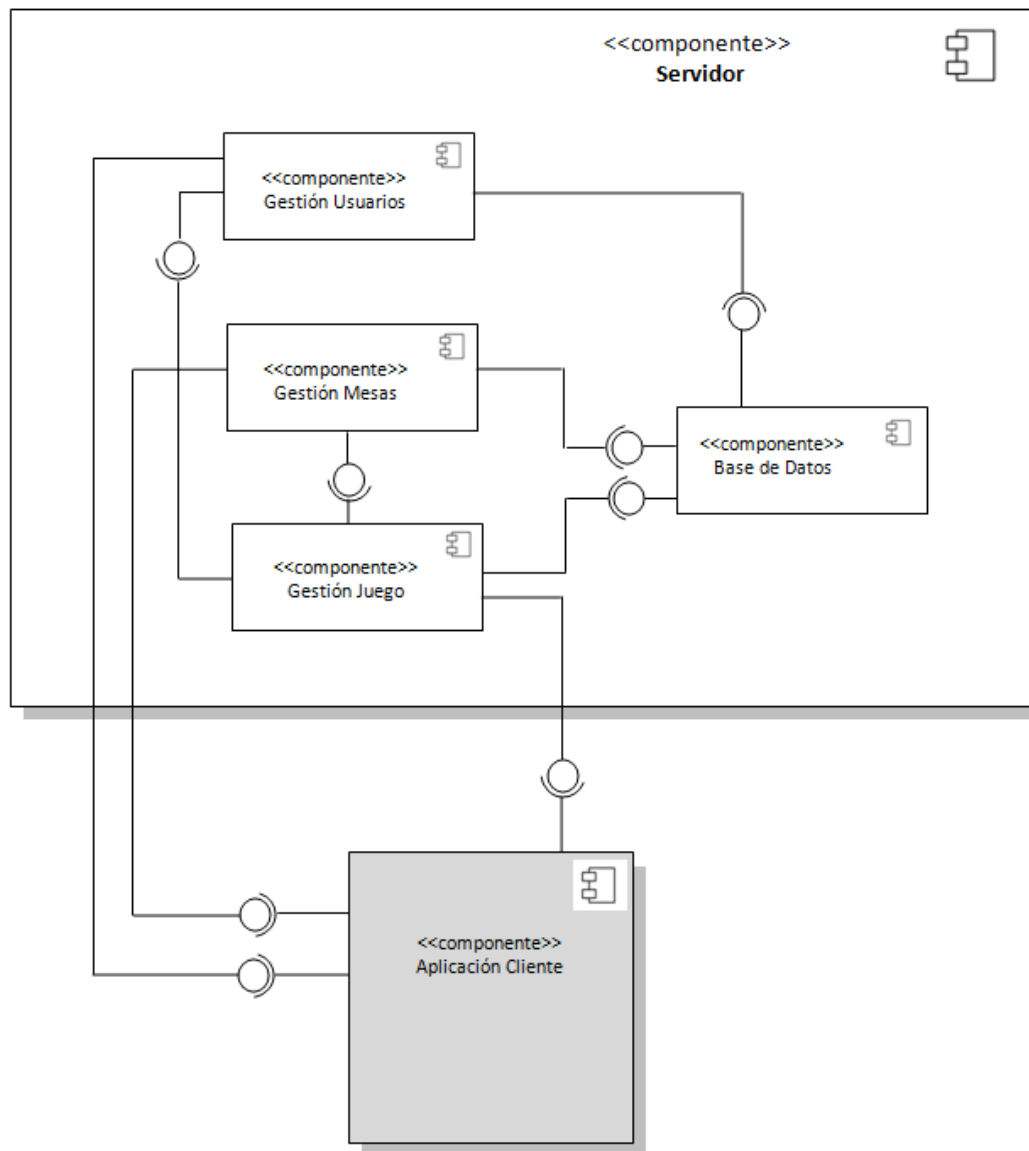


Figura 3 – Arquitectura preliminar detallada del sistema

3.4 Estudio tecnológico

En este apartado se especifica el estudio de diferentes tecnologías que se pueden aplicar para el desarrollo de la aplicación.

En las siguientes subsecciones se analizarán las tecnologías aplicables a los distintos componentes que forman parte del servidor. Las tecnologías que forman parte del cliente quedan fuera del alcance de este documento.

3.4.1 Tecnologías impuestas

Para la realización de este proyecto fin de carrera no se ha detectado ninguna tecnología que venga impuesta, ya que existen varias opciones en cuanto al desarrollo de una aplicación servidor.

3.4.2 Tecnologías aplicables al componente Base de Datos

En este apartado se muestra el estudio de las diferentes tecnologías analizadas dentro de las aplicables para el desarrollo de la base de datos del servidor.

El sistema a desarrollar necesitará almacenar, organizar y manejar información. Para poder facilitar las tareas de gestión, control y administración de esta información, los datos estarán centralizados en el nodo servidor. Si cada cliente tuviese una base de datos independiente, estas tareas serían demasiado tediosas.

Dentro de los sistemas gestores de bases de datos disponibles en el mercado se ha hecho una preselección de tres de los más populares y utilizados: Mysql [9], Oracle [10] y Sql Server [11].

La primera de las tecnologías estudiadas es el sistema gestor de base de datos Mysql, un sistema gestor de bases de datos relacional, que soporta multitud de sistemas operativos y es de código abierto y aunque existen versiones comerciales, Mysql puede usarse de manera gratuita siempre y cuando no se incluya el gestor dentro de un software propietario [12]. Además, existen multitud de lenguajes de programación que pueden

interactuar con este sistema, entre ellos Java [13], PHP [14], PERL [15] y Python [16]. Es fácilmente instalable y configurable. Suele utilizarse en combinación con Apache HTTP Server [17] y los lenguajes PERL, PHP y Python.

Otra de las opciones estudiadas, es Oracle. Consiste en un sistema gestor de base de datos relacional y multiplataforma [18]. Al igual que con Mysql, hay una gran cantidad de lenguajes de programación que pueden trabajar con Oracle. La principal diferencia con el sistema anterior es que éste no es de código abierto.

La última opción estudiada es Sql Server. Al igual que los anteriores, Sql Server es un sistema que trabaja con bases de datos relacionales [19]. Sin embargo, solo puede ser instalado en plataformas Windows, aunque puede ser accesible por aplicaciones que estén funcionando en cualquier plataforma. Al contrario que MySql, no es de código abierto.

3.4.3 Tecnologías aplicables a los componentes Gestión Usuarios, Gestión Mesas y Gestión Juego

En esta sección se describen las tecnologías aplicadas para los diferentes componentes que forman parte del servidor.

La principal funcionalidad de estos componentes es comunicarse con los nodos clientes, interactuar con la base de datos y manejar la lógica de negocio del sistema.

Para la programación de estos componentes se han estudiado tres de las tecnologías más usadas en la programación de componentes en el nodo servidor: JSP [20], PERL y PHP.

La primera tecnología estudiada es JSP (Java Server Page). Es una tecnología orientada a la creación de aplicaciones web dinámicas utilizando la programación en Java. Por lo que se trata de una tecnología orientada a objetos, multiplataforma y que puede usarse en multitud de servidores [21]. Además, se beneficia de la extendida comunidad de desarrolladores Java existente.

La segunda opción analizada es PERL, un lenguaje de programación de código abierto destinado a la programación de aplicaciones web dinámicas y siendo muy utilizado para la extracción y reporte de información. PERL está inspirado en diferentes lenguajes como C, sh y awk y permite la utilización de múltiples paradigmas de programación entre ellos

la orientación a objetos [22]. Aunque se trata de un lenguaje muy asociado a UNIX puede usarse en otras plataformas. Por lo general, es más fácil de aprender que JSP.

La tercera y última opción analizada es PHP, un lenguaje de programación de código abierto, flexible, multiplataforma y muy utilizado en la programación de código del lado del servidor. Aunque no nació como un lenguaje orientado a objetos, nos ofrece la posibilidad de poder utilizar este paradigma de programación [23]. Además, puede interactuar con la mayoría de sistemas gestores de bases de datos y servidores web que se utilizan hoy día. Dispone de gran cantidad de documentación y funciones predefinidas. De los tres lenguajes estudiados, PHP es el lenguaje que tiene la curva de aprendizaje más baja.

3.4.4 Tecnologías asociadas al servidor

En este punto se muestra el estudio de diferentes servidores web que se pueden utilizar para la implementación del sistema.

Dentro de todos los servidores que pueden ser utilizados se ha realizado un estudio de dos de los servidores web más populares y utilizados: el servidor Apache HTTP Server y el servidor IIS [24].

El primero de los servidores analizados es Apache HTTP Server, más conocido como Apache, es un servidor web HTTP de código abierto y gratuito, multiplataforma y que permite emplear diferentes lenguajes de programación tales como PHP, PERL y Python. Permite conexiones seguros a través del protocolo HTTPS. Destaca por su robustez, seguridad y rendimiento [25].

La otra opción estudiada es IIS, el acrónimo de Internet Information Services. Es un servidor web propiedad de Microsoft que ofrece múltiples servicios entre ellos: FTP, SMTP, NNTP, HTTP y HTTPS. Al igual que Apache, admite varios lenguajes de programación, pero solo puede ser usado en plataformas Windows.

3.5 Selección de tecnologías

En este apartado se exponen las diferentes tecnologías seleccionadas para el desarrollo del sistema, así como el razonamiento de tales elecciones.

De todas las tecnologías aplicables y estudiadas en la sección 3.3 de este documento, se ha decidido escoger una tecnología de cada uno de los puntos tratados con el fin de simplificar y facilitar el desarrollo del sistema y la integración de los diferentes componentes.

De los diferentes sistemas gestores de bases de datos estudiados y analizados se ha decidido trabajar con Mysql. Se ha elegido este sistema gestor de bases de datos por tratarse de código abierto, por ser multiplataforma, por su interacción con los lenguajes de programación y servidores más populares y utilizados. Además, es fácilmente instalable y configurable y sencillo de utilizar.

Para programar la lógica del sistema se ha decidido utilizar el lenguaje de programación PHP. Uno de los motivos principales por los que se ha elegido este lenguaje es por su interacción con Mysql. Otros de los motivos que ha llevado a esta decisión es que se trata de un lenguaje fácil de aprender, de código abierto, flexible, multiplataforma y muy utilizado en el lado del servidor.

De los diferentes servidores estudiados se ha seleccionado Apache HTTP Server. Los motivos que han propiciado esta elección son que se trata de un servidor libre y de código abierto, multiplataforma y fácilmente integrable con Mysql y PHP. Concretamente, se ha decidido utilizar la distribución de Apache Xampp [26]. Se trata de una distribución completamente gratuita, multiplataforma, fácil de instalar y manejar e incluye Mysql, PHP y PERL.

3.6 Arquitectura definitiva de alto nivel

Partiendo de la arquitectura especificada en la sección 3.3 y teniendo en consideración la decisión tomada en cuanto a las tecnologías a utilizar hay que añadir a la arquitectura cliente-servidor el componente Google Cloud Messaging (GCM [27]). Este componente será el que se encargue de la sincronización de los usuarios dentro de una partida. La siguiente figura muestra la arquitectura definitiva:

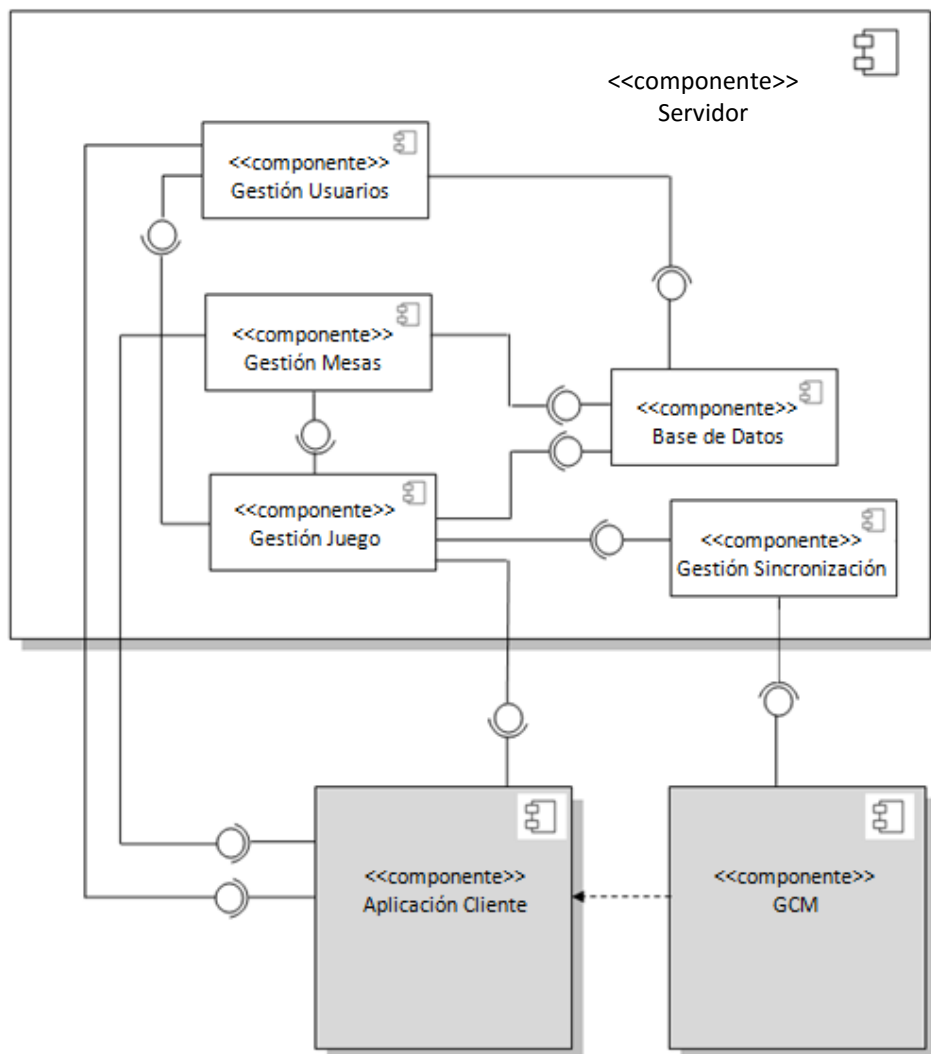


Figura 4 – Arquitectura definitiva de alto nivel

Google Cloud Messaging está formado por un conjunto de servidores en la nube que proveen un servicio gratuito de envío de mensajes que hará de intermediario entre los

clientes y el servidor, cuando este último quiera mandar un mensaje a diferentes dispositivos sin que estos lo soliciten favoreciendo la sincronización entre los mismos.

Este servicio permite que la aplicación pueda registrar un dispositivo móvil para recibir mensajes, obteniendo un identificador. Este identificador es enviado por la aplicación al servidor, donde se almacena. Cuando el servidor lo necesite, podrá mandar un mensaje a los servidores de GCM, el cual reenviará el mensaje al dispositivo o dispositivos registrados anteriormente en los servidores de Google Cloud Messaging.

Las ventajas de esta tecnología son las siguientes:

- No necesita que los clientes soliciten nada para poder enviar mensajes a éstos desde el servidor.
- No es necesario que la aplicación permanezca ejecutándose, sino que el sistema la despertará cuando reciba un mensaje, permitiendo un mejor consumo de recursos y de batería del Smartphone.

Como puntos en contra cabe destacar que:

- Requiere dispositivos con Android 2.2 o con la aplicación de Google Play.
- Es necesario que los usuarios tengan configurada una cuenta de Google en el dispositivo.

Estas desventajas son casi inapreciables debido a que la gran mayoría de dispositivos Android que están en uso hoy en día ya tienen una versión superior y si deseas bajarte cualquier aplicación del Play Store es necesaria tener una cuenta, por lo que también en la casi totalidad de los casos esto no afecta.

3.7 Casos de uso

En este capítulo se muestra el diagrama de casos de uso de la aplicación junto con la definición de cada uno de ellos. El diagrama de casos de uso nos sirve para describir que hace el sistema. El estudio de este diagrama facilitará la identificación de requisitos funcionales.

3.7.1 Diagramas de casos de uso

La siguiente figura muestra el diagrama de los casos de uso identificados para el servidor:

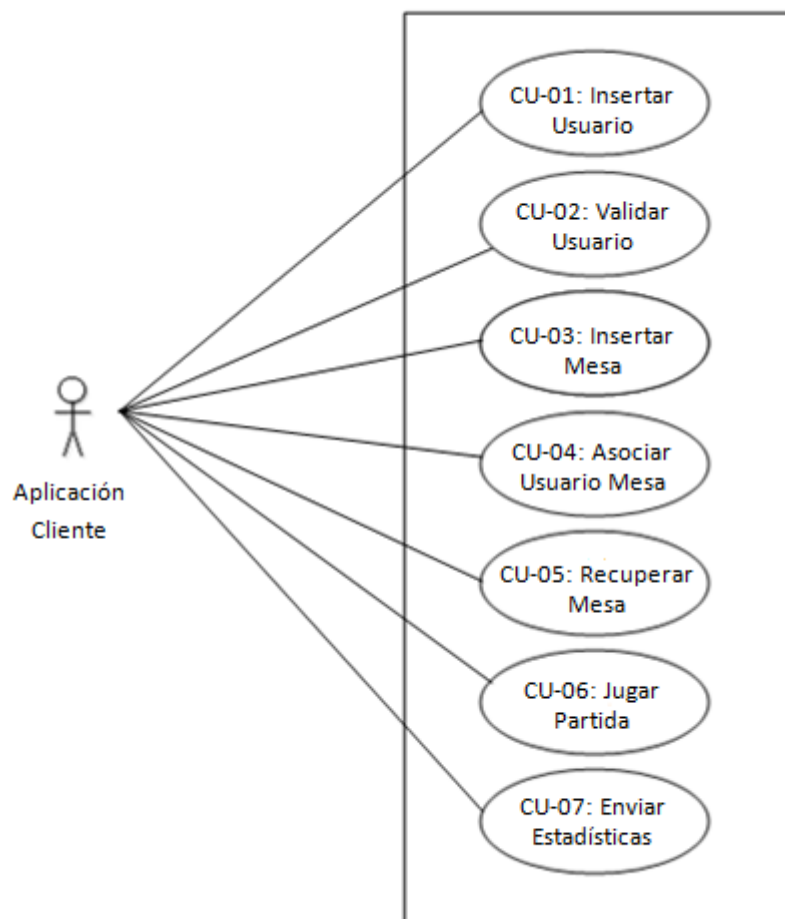


Figura 5 – Diagrama de casos de uso

3.7.2 Definición textual de los casos de uso

En este apartado se muestra la definición textual de cada uno de los casos de uso identificados en el diagrama del apartado anterior. Para presentar esta información se utilizará el formato definido en la plantilla 1 del Anexo 3.

En la siguiente tabla se describe el primer caso de uso identificado. Éste consiste en la inserción de un usuario en el sistema de a partir de la información proporcionada por la aplicación cliente.

CU-01	
Nombre:	Insertar Usuario
Descripción:	Insertar un nuevo usuario en el sistema a partir de la información proporcionada por la aplicación cliente.
Actores:	Aplicación cliente.
Precondiciones:	Servidor arrancado y usuario no existe en el sistema.
Postcondiciones:	Se informa a la aplicación cliente del estado de la inserción.
Flujo normal:	1. Aplicación solicita la operación de insertar usuario. 2. Se comprueba si existe el usuario a insertar. 3. Se devuelve la confirmación de inserción a la aplicación cliente.
Flujo Alternativo:	

Tabla 1 - Definición textual del caso de uso CU-01

En la tabla que se muestra a continuación se especifica el segundo caso de uso definido consistente en la validación de un usuario. El servidor deberá comprobar que los datos proporcionados por la aplicación cliente son correctos y que se corresponden con un usuario previamente insertado.

CU-02	
Nombre:	Validar Usuario
Descripción:	Validar la información proporcionada referente a un usuario insertado anteriormente en el sistema a partir de los datos proporcionados por la aplicación cliente.
Actores:	Aplicación cliente.
Precondiciones:	Servidor arrancado y usuario existente en el sistema.
Postcondiciones:	Se informa a la aplicación cliente del estado de la validación
Flujo normal:	<ol style="list-style-type: none"> 1. Aplicación cliente solicita la operación de validar usuario. 2. Se comprueba si la información proporcionada por el cliente existe en el sistema. 3. Se devuelve estado de la validación a la aplicación cliente.
Flujo Alternativo:	

Tabla 2 - Definición textual del caso de uso CU-02

En la tabla de abajo se define el tercer caso de uso identificado. Este caso de uso consiste en la inserción de una mesa en el sistema con los datos provistos por la aplicación cliente.

CU-03	
Nombre:	Insertar Mesa
Descripción:	Insertar una nueva mesa partir de la información proporcionada por la aplicación cliente.
Actores:	Aplicación cliente.
Precondiciones:	Servidor arrancado y mesa no existente en el sistema.
Postcondiciones:	Se informa a la aplicación cliente del estado de la inserción.
Flujo normal:	<ol style="list-style-type: none"> 1. Aplicación solicita la operación de insertar usuario. 2. Se comprueba si existe la mesa a insertar. 3. Se devuelve la confirmación de inserción a la aplicación cliente.
Flujo Alternativo:	

Tabla 3 - Definición textual del caso de uso CU-03

A continuación, se muestra la tabla con la descripción del caso de uso referente a la asociación de un usuario a una mesa. El servidor deberá poder asociar un usuario previamente logado en el sistema desde la aplicación cliente a una mesa anteriormente insertada.

CU-04	
Nombre:	Asociar Usuario Mesa
Descripción:	Asociar un jugador a una mesa.
Actores:	Aplicación cliente.
Precondiciones:	Servidor arrancado y mesa y usuario existentes en el sistema.
Postcondiciones:	Se informa a la aplicación cliente del estado de la asociación y se le asigna al usuario un número de posición dentro de la mesa.
Flujo normal:	<ol style="list-style-type: none"> 1. Aplicación cliente solicita la operación de validar usuario. 2. Se comprueba si los datos proporcionados por el usuario referentes a la mesa existen en el sistema. 3. La aplicación cliente solicita el número de usuarios que hay en la mesa proporcionada. 4. El servidor responde con el número de usuarios que hay en la mesa. 5. La aplicación cliente comprueba que queda espacio en la mesa y solicita la asociación del usuario. 6. Se realiza la asociación del usuario a la mesa por parte del servidor. 7. Se envía mensaje de confirmación de la asociación a la aplicación cliente.
Flujo Alternativo:	

Tabla 4 - Definición textual del caso de uso CU-04

En la siguiente tabla se especifica el caso de uso referente a la recuperación de una mesa. El servidor deberá poder recuperar toda la información asociada una mesa previamente insertada.

CU-05	
Nombre:	Recuperar Mesa
Descripción:	Recuperar la información de a una mesa asociada a un jugador.
Actores:	Aplicación cliente.
Precondiciones:	Servidor arrancado y usuario y mesa existentes en el sistema.
Postcondiciones:	Se informa a la aplicación cliente del estado de la recuperación y se recupera la información.
Flujo normal:	<ol style="list-style-type: none"> 1. Aplicación cliente solicita la operación de recuperar mesa. 2. Se comprueba que los datos proporcionados por la aplicación cliente respecto a la mesa son correctos. 3. Se comprueba que el usuario suministrado por la aplicación cliente está asociado a la mesa. 4. Se recupera la información de la mesa. 5. Se informa a la aplicación cliente de la recuperación.
Flujo Alternativo:	

Tabla 5 - Definición textual del caso de uso CU-05

En la tabla situada debajo se define el siguiente caso de uso. Este caso de uso consiste en la gestión y control de una partida de Cuatrola.

CU-06	
Nombre:	Jugar Partida
Descripción:	Gestionar una partida de cuatrola
Actores:	Aplicación cliente.
Precondiciones:	Servidor arrancado y mesa existente en el sistema con cuatro usuarios asociados.
Postcondiciones:	Se gestiona y controla una partida de Cuatrola.
Flujo normal:	1. Aplicación cliente solicita el inicio de una partida. 2. Se juega la partida de Cuatrola.
Flujo Alternativo:	

Tabla 6 - Definición textual del caso de uso CU-06

Por último, la tabla siguiente muestra el caso de uso relativo al envío de las estadísticas asociadas a un usuario. El servidor deberá poder consultar las estadísticas referentes a un usuario y enviárselas a la aplicación cliente.

CU-07	
Nombre:	Enviar Estadísticas
Descripción:	Obtener la información referente a las estadísticas de partidas jugadas y jugadas realizadas de un usuario y enviárselas a la aplicación cliente.
Actores:	Aplicación cliente.
Precondiciones:	Servidor arrancado.
Postcondiciones:	Se envía a la aplicación cliente la información solicitada..
Flujo normal:	1. Aplicación cliente solicita la información de las estadísticas de un usuario. 2. Se obtiene la información de las estadísticas de un usuario. 3. Se envía la información a la aplicación cliente.
Flujo Alternativo:	

Tabla 7 - Definición textual del caso de uso CU-07

3.8 Requisitos de software

En apartado se presentan los requisitos de software identificados que tendrán que ser cubiertos para cumplir con los objetivos especificados para este proyecto fin de carrera. A continuación se muestran los tipos de requisitos funcionales y no funcionales.

3.8.1 Requisitos funcionales

La siguiente tabla muestra la información de los requisitos funcionales que debe cumplir la parte servidor del sistema de acuerdo al formato definido en la plantilla 2 del Anexo 3.

Requisitos funcionales				
Id	Nombre	Descripción	Necesidad	Prioridad
RF-01	Insertar usuario	Insertar un nuevo usuario en el sistema con los datos proporcionados por la aplicación cliente.	Alta	Alta
RF-02	Validar usuario	Comprobar que los datos proporcionados por la aplicación cliente referentes a un usuario existen en el sistema y son correctos.	Alta	Media
RF-03	Insertar mesa	Insertar una nueva mesa en el sistema con los datos proporcionados por la aplicación cliente	Alta	Alta
RF-04	Validar mesa	Comprobar que los datos proporcionados por la aplicación cliente referentes a una mesa existen en el sistema y son correctos.	Alta	Media
RF-05	Asociar un usuario a una mesa	Asociar un usuario existente en el sistema a una mesa anteriormente insertada.	Alta	Alta
RF-06	Recuperar partida	Recuperar la información referente a una mesa existente en el sistema.	Media	Baja
RF-07	Comprobar usuario asociado a mesa	Comprobar que un usuario está asociado a una mesa.	Media	Baja
RF-08	Comprobar jugadores mesa	Comprobar el número de jugadores en una mesa.	Alta	Alta
RF-09	Barajar cartas	Barajar las cartas a repartir	Alta	Alta
RF-10	Repartir cartas	Repartir cartas a los diferentes usuarios.	Alta	Alta
RF-11	Comprobar cartas usuario	Comprobar que cartas puede lanzar un usuario.	Alta	Alta
RF-12	Tirar cartas	Actualizar en la mesa la carta lanzada por el usuario.	Alta	Alta
RF-13	Actualizar jugada	Actualizar en el sistema la jugada elegida por un usuario.	Alta	Alta

Requisitos funcionales				
Id	Nombre	Descripción	Necesidad	Prioridad
RF-14	Comprobar jugada mayor	Comprobar cuál es la jugada más alta de las elegidas por los distintos usuarios.	Alta	Alta
RF-15	Comprobar cante 20	Comprobar si un usuario puede cantar las 20.	Alta	Media
RF-16	Comprobar cante 40	Comprobar si un usuario puede cantar las 40.	Alta	Media
RF-17	Comprobar carta ganadora	Comprobar que carta de las lanzadas por los diferentes usuarios es la ganadora.	Alta	Alta
RF-18	Calcular puntos baza	Calcular los puntos realizados por cada pareja en una baza.	Alta	Alta
RF-19	Obtener turno	Obtener el nuevo turno tras finalizar una baza.	Alta	Media
RF-20	Comprobar pareja ganadora	Comprobar que pareja gana una mano.	Alta	Alta
RF-21	Calcular puntos globales	Calcular los puntos globales de cada pareja.	Alta	Alta
RF-22	Calcular estadísticas	Calcular las estadísticas de un usuario mientras juega una partida.	Media	Baja
RF-23	Obtener estadísticas	Obtener las estadísticas de un usuario determinado.	Media	Baja
RF-24	Comprobar fin partida	Comprobar si una pareja ha alcanzado la máxima puntuación.	Alta	Alta
RF-25	Obtener ganadores partida	Comprobar que usuarios ganan la partida.	Alta	Alta

Tabla 8 – Requisitos funcionales

3.8.2 Requisitos no funcionales

En la Tabla 9 se describe la información de los distintos requisitos no funcionales recogidos, siguiendo el formato definido en la plantilla 2 del Anexo 3.

Requisitos no funcionales				
Id	Nombre	Descripción	Necesidad	Prioridad
RNF-01	Comunicación con aplicaciones cliente	Comunicar el servidor con las diferentes aplicaciones cliente.	Alta	Alta
RNF-02	Comunicación con GCM	Comunicar el servidor con Google Cloud Messaging.	Alta	Alta
RNF-02	Capacidad	La aplicación debe soportar un número elevado de usuarios conectados y jugando a la vez.	Media	Baja
RNF-03	Tiempo de respuesta	El tiempo de respuesta de la aplicación no debe exceder los 30 segundos.	Media	Media
RNF-04	Mantenibilidad	La parte servidor de la aplicación debe ser fácilmente mantenerle.	Media	Media
RNF-05	Disponibilidad	El servidor debe estar accesible 24 horas al día.	Alta	Alta
RNF-06	Portable	La parte servidor de la aplicación debe ser portable.	Media	Media

Tabla 9 – Requisitos no funcionales

3.9 Diseño del plan de pruebas de aceptación

En esta sección se detalla el plan de pruebas de aceptación del sistema, en donde se muestran las distintas pruebas que se han de superar y los resultados esperados de las mismas para que se superen estas pruebas.

La Tabla 10 detalla las pruebas de aceptación definidas de acuerdo al formato establecido en la plantilla 3 del Anexo 3.

Pruebas de aceptación			
Id	Requisitos probados	Entrada	Salida
PA-01	RF-01	Insertar un usuario recibiendo el nombre de usuario y la contraseña de la aplicación cliente.	Se inserta el usuario correctamente en el sistema y se envía la confirmación a la aplicación cliente.
PA-02	RF-02	Validar un usuario existente en el sistema recibiendo el nombre de usuario y la contraseña de la aplicación cliente.	Se valida el usuario con sus datos correctamente y se envía la confirmación a la aplicación cliente.
PA-03	RF-03	Insertar una mesa recibiendo el nombre de mesa y la contraseña de la aplicación cliente.	Se inserta la mesa correctamente en el sistema y se envía la confirmación a la aplicación cliente.
PA-04	RF-04	Validar una mesa existente en el sistema recibiendo el nombre de mesa y la contraseña de la aplicación cliente.	Se valida la mesa con sus datos correctamente y se envía la confirmación a la aplicación cliente.
PA-05	RF-05, RF-08	Asociar un usuario a una mesa existente recibiendo el nombre de usuario y los datos de la mesa de la asociación cliente.	Se asocia el usuario en una posición libre de la mesa y se comunica a la aplicación cliente el resultado de la asociación.
PA-06	RF-05, RF-06, RF-07, RF-08	Recuperar una partida existente en una mesa recibiendo el nombre de usuario y los datos de la mesa de la aplicación cliente.	Se recupera la información de la partida y se envía la información a la aplicación cliente.
PA-07	RF-09, RF-10	Se recibe la solicitud de la aplicación cliente de iniciar partida.	Se barajan y se reparten las cartas a los usuarios de la mesa.
PA-08	RF-11	Se comprueban las cartas de un usuario recibiendo desde la aplicación cliente la posición de éste dentro de la mesa.	Se comprueba que cartas se pueden lanzar el usuario y se envía la respuesta a la aplicación cliente.
PA-09	RF-12	Se recibe desde la aplicación cliente la posición del usuario dentro de la mesa y la carta a lanzar.	Se actualiza la mesa con la carta lanzada por el usuario y se comunica a las aplicaciones cliente que se ha realizado un lanzamiento.
PA-10	RF-13	Se recibe desde la aplicación cliente la posición del usuario en la mesa y la jugada elegida.	Se actualiza los datos de la mesa con la jugada elegida por el usuario
PA-11	RF-14	Se recibe la solicitud de la aplicación cliente de comprobar cuál es la jugada mayor.	Se comprueba que jugada de las elegidas por los usuarios es la más alta y se comunica a las diferentes aplicaciones clientes.
PA-12	RF-15	Se recibe la solicitud de comprobar cante 20 recibiendo la posición del usuario que quiere cantar.	Se comprueba si el usuario puede realizar un cante de 20 con las cartas que posee y se comunica a las aplicaciones cliente.
PA-13	RF-16	Se recibe la solicitud de comprobar cante 40 recibiendo la posición del usuario que quiere cantar.	Se comprueba si el usuario puede realizar un cante de 40 con las cartas que posee y se comunica a las aplicaciones cliente.

Pruebas de aceptación			
Id	Requisitos probados	Entrada	Salida
PA-14	RF-17, RF18, RF-19	Se finaliza una baza.	Se comprueba que carta y jugador ganan la baza, se calculan los puntos del jugador ganador, se incrementa el marcador de baza de la pareja y se calcula el próximo turno. Se comunica el fin de la baza a las aplicaciones cliente.
PA-15	RF-20, RF-21	Se finaliza una mano.	Se comprueba que pareja gana la mano y se incrementa su marcador global. Se comunica el fin de la mano a las aplicaciones cliente.
PA-16	RF-20, RF-21, RF-22	Uno de los usuarios elige una jugada diferente de normal y se juega una mano.	Se comprueba que pareja gana la mano, se incrementa su marcador global y se actualizan sus estadísticas. Se comunica el fin de la mano a las aplicaciones cliente.
PA-17	RF-22, RF-24, RF-25	Se da una partida por finalizada.	Se comprueba el fin de la partida, se obtiene la pareja ganadora y se actualizan sus estadísticas. Se comunica el fin de partida y la pareja ganadora a las aplicaciones cliente.
PA-18	RF-26	Se solicita desde la aplicación cliente la obtención de estadísticas recibiendo el nombre de usuario.	Se envía a la aplicación cliente las estadísticas del usuario.

Tabla 10 – Pruebas de aceptación

Capítulo 4

Diseño Detallado

4.1 Diseño de Software

En este apartado se hará una descripción detallada de los diferentes componentes que conforman el sistema identificados en el capítulo de análisis.

En la siguiente figura se muestra la arquitectura definitiva del sistema indicando los apartados en los que se describirán cada uno de los diferentes componentes:

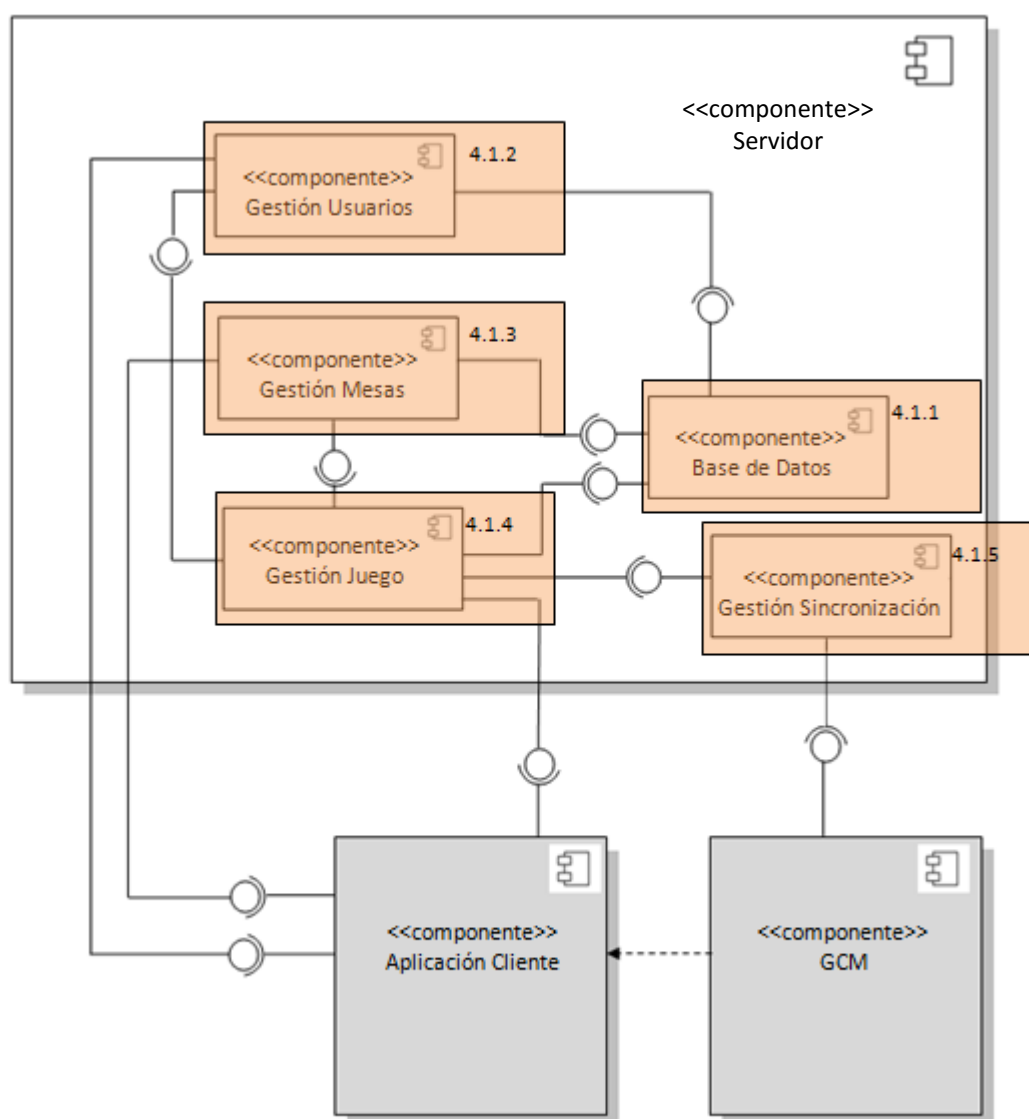


Figura 6 - Diagrama de componentes definitivo

En este documento solo se detallarán los componentes concernientes al servidor, los componentes de la aplicación cliente y GCM quedan fuera del alcance de este proyecto.

4.1.1 Componente Base de datos

La funcionalidad de este componente se centra en almacenar, organizar y manejar la información necesaria para el correcto funcionamiento del sistema a desarrollar. Esta información podrá ser recuperada, creada o modificada por diferentes componentes.

En la figura 7 se muestra el modelo relacional concerniente a la base de datos:

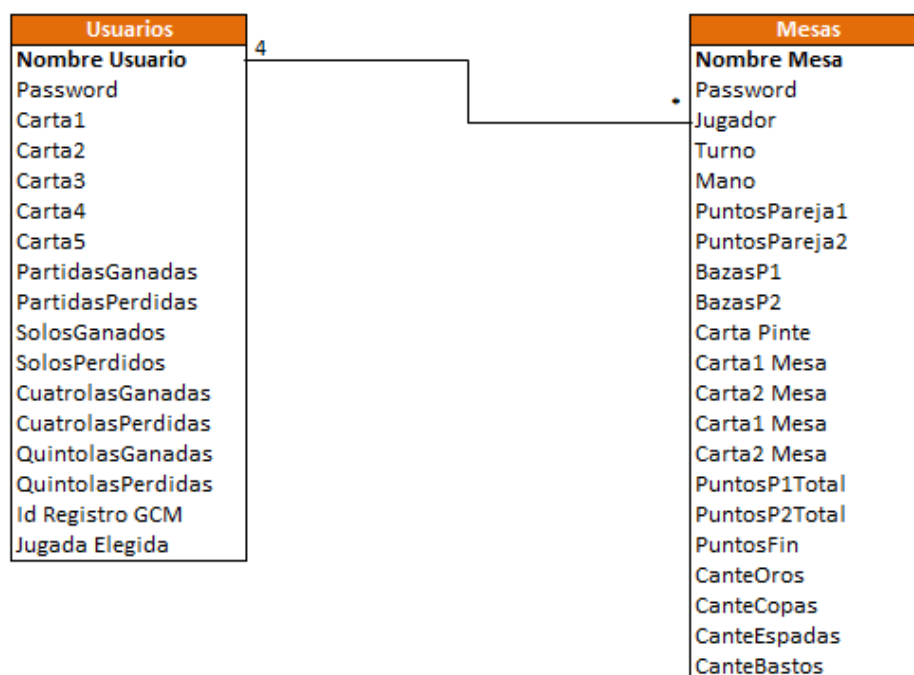


Figura 7 – Modelo relacional de la base de datos

Como se puede ver en la figura anterior la base de datos está formada por dos tablas: usuarios y mesas.

La tabla usuarios es la encargada de almacenar, organizar y manejar la información de los usuarios. Esta tabla estará formada por los campos nombre de usuario, contraseña, los campos correspondientes a las cartas del jugador y a la jugada elegida, los campos encargados de guardar la información en cuanto a partidas y jugadas realizadas por el

usuario y el campo identificador de registro, donde se almacenará el número de registro de GCM del dispositivo móvil del usuario.

La tabla mesas es la que se ocupa de almacenar, organizar y manejar la información de las diferentes mesas, así como todos los datos necesarios para poder jugarse una partida dentro de la mesa. Esta tabla estará compuesta por los campos nombre de mesa, contraseña, los diferentes jugadores que componen la mesa y todos los campos necesarios para poder gestionar una partida completa. Además, cada vez que se termine una partida, el registro referente a dicha partida será eliminado.

4.1.2 Componente Gestión de Usuarios

Este componente se encargará de gestionar los usuarios que accedan al sistema. Por ello, este componente tendrá las siguientes funciones principalmente: registrar, validar y obtener y enviar las estadísticas de cada usuario.

La Figura 8 muestra el diagrama de clases que forman este componente:

Usuario
+nombre: String -contraseña: String +partidasganadas : int +partidasperdidas : int +solosganados : int +solosperdidos : int +cuatrolaganadas : int +cuatrolasperdidas : int +quintolasganadas : int +quintolasperdidas: int +identificadorregistro: String
-crearUsuario (nombre, contraseña : String): Usuario -comprobarUsuario (nombre: String): boolean +validarAccesoUsuario(nombre, contraseña : String) : boolean +obtenerSolosGanados() : int +obtenerSolosPerdidos() : int +obtenerCuatrolasGanadas(): int +obtenerCuatrolasPerdidas(): int +obtenerQuintolasGanadas(): int +obtenerQuintolasPerdidas(): int +obtenerEstadísticas(): int[8] +incrementarSolosGanados() +incrementarSolosPerdidos() +incrementarCuatrolasGanadas() +incrementarCuatrolasPerdidas() +incrementarQuintolasGanadas() +incrementarQuintolasPerdidas () +obtenerIdentificadorRegistro(): String +actualizarIdentificadorRegistro(idregistro:String)

Figura 8 – Modelo de clases del componente Gestión de Usuarios

Este componente está formado únicamente por la clase Usuario. Recibirá la información referente a nombre usuario, contraseña e identificador de registro del componente aplicación cliente y almacenará y obtendrá la información requerida del componente base datos.

4.1.3 Componente Gestión de Mesas

El componente Gestión de Mesas se ocupará de gestionar las mesas que se generen en el sistema. Por ello, realizará las siguientes operaciones: registrar y validar mesas, asociar usuarios a las mesas y asignarles una posición libre dentro de la misma.

La Figura 9 muestra el diagrama de clases que forman este componente.

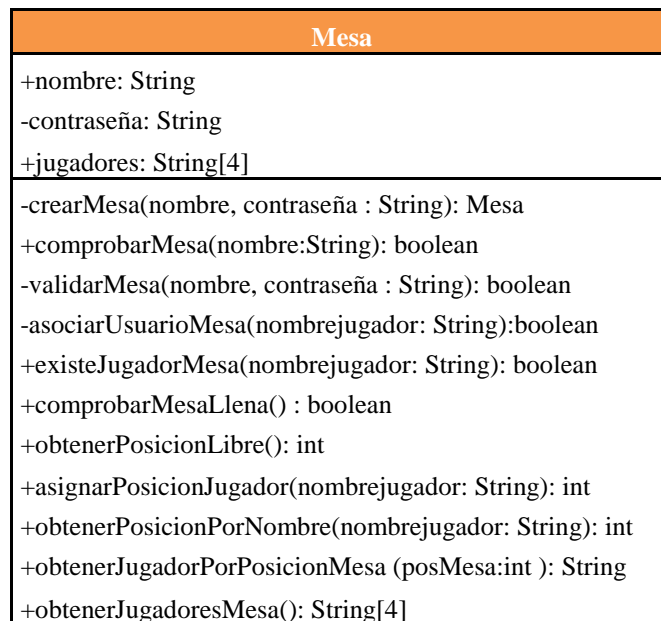


Figura 9 – Modelo de clases del componente Gestión de Mesas

Como se puede ver en el diagrama de arriba, la clase Mesa está formada, entre otros atributos, por cuatro jugadores cuyos nombres coinciden con el de alguno de los usuarios dados de alta en el sistema. Esta clase estará asociada con la clase Usuarios del componente descrito en el apartado anterior. Este componente recibirá la información del componente cliente, almacenando u obteniendo la información solicitada en la base de datos.

4.1.4 Componente Gestión Juego

Este componente es el más importante de todos, y es el que se ocupará de controlar la lógica del juego de la cuatrola. Por lo tanto tendrá que encargarse de todas las operaciones necesarias para poder jugar una partida siguiendo las bases y normas del

juego, además de informar de las estadísticas que se van generando durante la partida. Entre sus funciones están: repartir y barajar las cartas, gestionar los turnos, comprobar que cartas pueden lanzarse, permitir a los jugadores tirar cartas, comprobar la carta ganadora en cada baza, permitir la selección de jugadas en cada mano, comprobar la jugada mayor, calcular los puntos de las bazas y manos, calcular los puntos totales, permitir a los jugadores cantar, comprobar el cante realizado, comprobar el fin de la partida y comprobar el ganador de la misma.

En la Figura 10 se muestra el diagrama de clases que forman este componente:



Figura 10 – Modelo de clases del componente Gestión del Juego

Como puede verse en la figura anterior, este componente está formado por las clases carta, jugador y partida Cuatrola.

Dentro de las clases que componen este componente, la principal y la que lleva el peso de la lógica del juego es la clase partida. Esta clase con sus métodos y atributos y con la ayuda de las clases jugador y carta gestionará una partida de Cuatrola.

4.1.5 Componente Gestión Sincronización

La funcionalidad de este componente consiste en conseguir que los cuatro jugadores de una mesa, estén totalmente sincronizados durante toda la partida. Por lo que este componente tiene que comunicarse con el componente GCM. Y para ello se usará la clase EnvíoMensajeGCM.

La siguiente figura muestra el diagrama de clases del componente GCM:

EnvíoMensajeGCM
+apikey: String
+idregistro: String[4]
+message: String
+urlgcm: String
+envioMensajeGCM(apikey, message, urlgcm: String, idregistro: String[4])
+enviarMensaje(): Boolean

Figura 11 – Modelo de clases del componente Gestión Sincronización

4.2 Diagramas de secuencia

Tras realizar el diseño de la aplicación servidor, se procede a reflejar las diferentes interacciones entre las aplicaciones cliente, el servidor y las distintas clases definidas en éste último en la ejecución de los casos de uso especificados en el apartado 3.7.

Con el fin de simplificar los diagramas y que éstos sean los más claros posible, se ha decidido no representar las interacciones con clases del lenguaje, así como las invocaciones a métodos que no aportan información relevante. Tampoco se van a representar los diagramas de secuencia de los flujos alternativos de los distintos casos de uso, mostrándose siempre el flujo de ejecución normal.

4.2.1 Insertar usuario (CU-01)

En la Figura 12 se especifica el diagrama de secuencia correspondiente al caso de uso de inserción de un usuario.

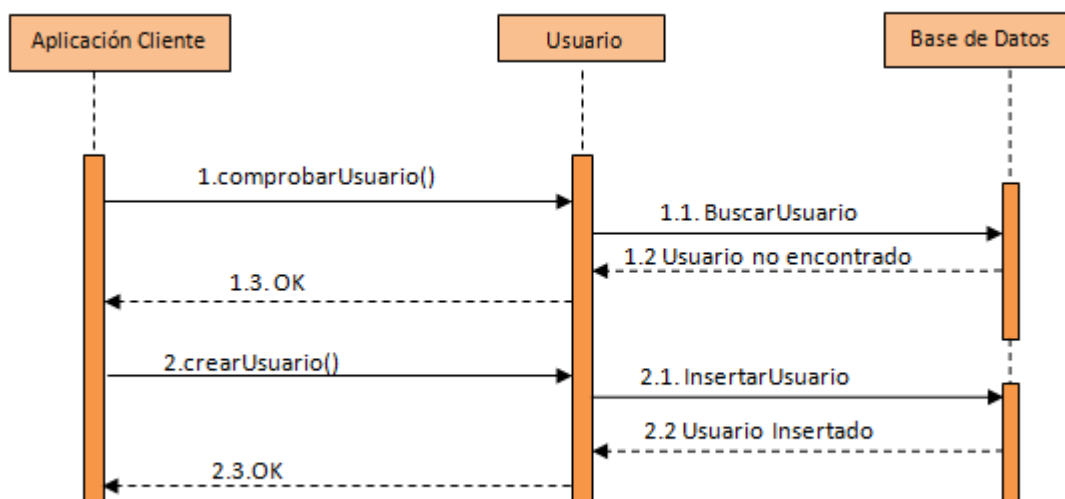


Figura 12 – Diagrama de secuencia de insertar usuario (CU-01)

Para insertar un usuario, el cliente le envía la petición de inserción al servidor con el nombre de usuario y la contraseña de acceso. Éste comprobará que el usuario no está

dado de alta en la base de datos y procederá a la inserción del mismo, devolviendo al final al cliente un mensaje de confirmación para indicar que el usuario se ha insertado.

4.2.2 Validar Usuario (CU-02)

La Figura 13 muestra el diagrama de secuencia que se corresponde con el caso de uso de validar usuario.

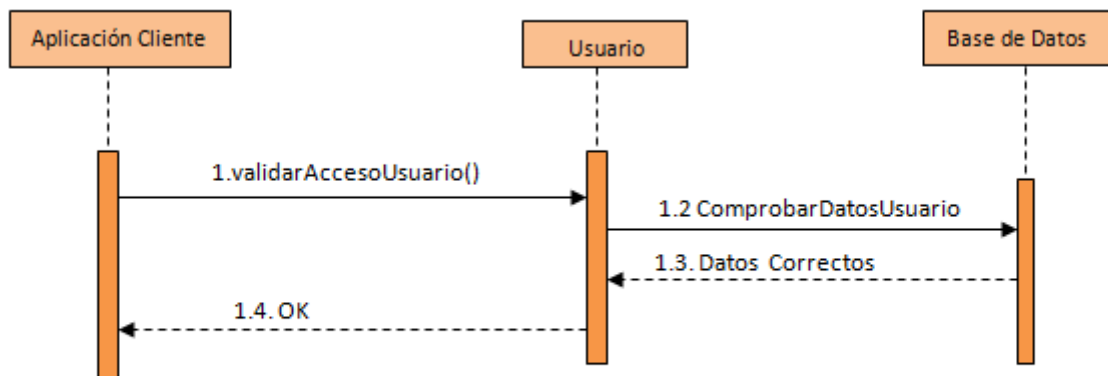


Figura 13 – Diagrama de secuencia de validar usuario (CU-02)

Para la validación de un usuario ya insertado, la aplicación cliente envía la solicitud de validación al servidor para que éste compruebe si existe el usuario y si la contraseña introducida es correcta. El servidor devolverá un mensaje con el estado de la validación.

4.2.3 Insertar Mesa (CU-03)

En este diagrama de secuencia se detallan las interacciones de la aplicación cliente con el servidor para insertar una mesa en el sistema.

En la Figura 14, se enseña el diagrama de secuencia correspondiente al caso de uso insertar mesa.

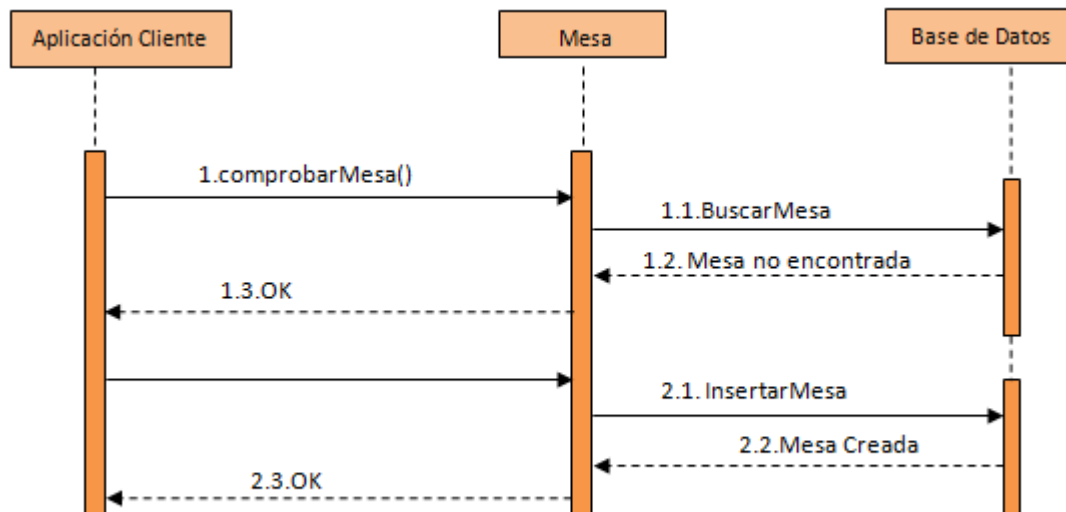


Figura 14 – Diagrama de secuencia de insertar mesa (CU-03)

Para insertar una mesa, la aplicación cliente envía la solicitud de inserción al servidor. Éste comprobará que la mesa no existe en la base de datos y continuará con la inserción de la misma, devolviendo al final a la aplicación cliente un mensaje de confirmación.

4.2.4 Asociar Usuario Mesa (CU-04)

La siguiente figura muestra el diagrama de secuencia del caso de uso referente a la asociación de un usuario a una mesa existente en el sistema:

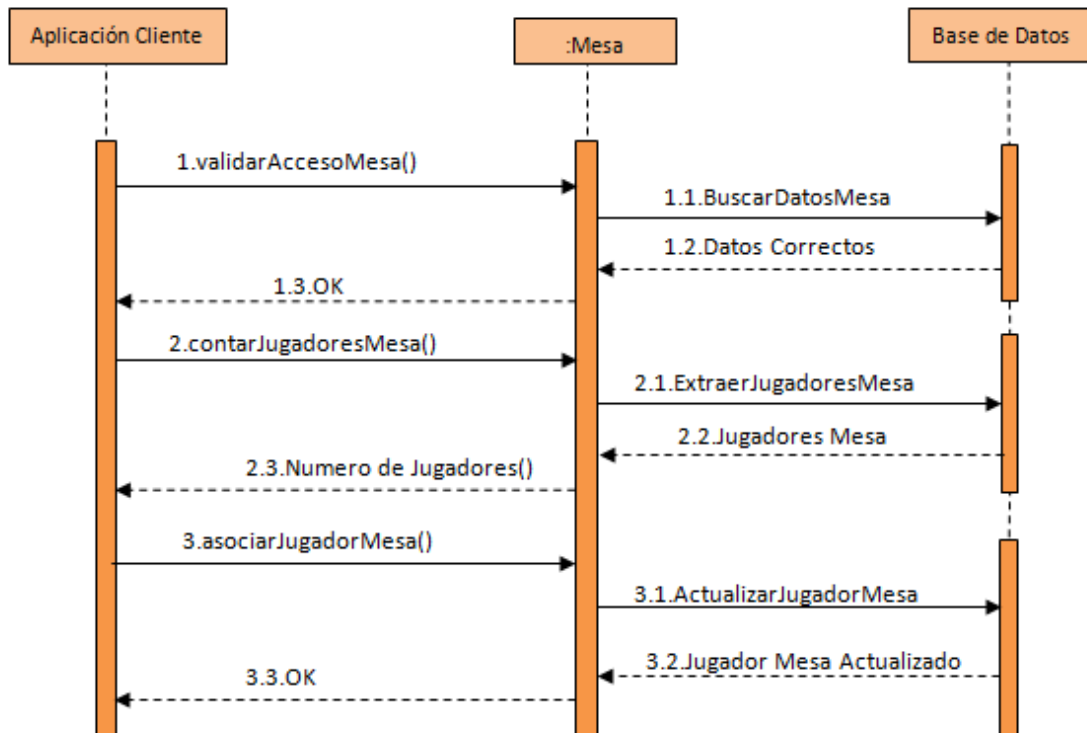


Figura 15 – Diagrama de secuencia de asociar usuario mesa (CU-04)

Para que la aplicación cliente pueda solicitar al servidor la asociación de un usuario a una mesa, se necesitará validar los datos de la mesa y comprobar que la mesa tenga menos de cuatro usuarios asociados. El servidor enviará al cliente el número de usuarios de una mesa y éste solicitará la asociación del usuario a la mesa, asignándole una posición vacía. Como respuesta final la aplicación cliente recibirá la confirmación de la asociación.

4.2.5 Recuperar partida (CU-05)

En la Figura 16 se muestra el diagrama de secuencia correspondiente al caso de uso recuperar partida.

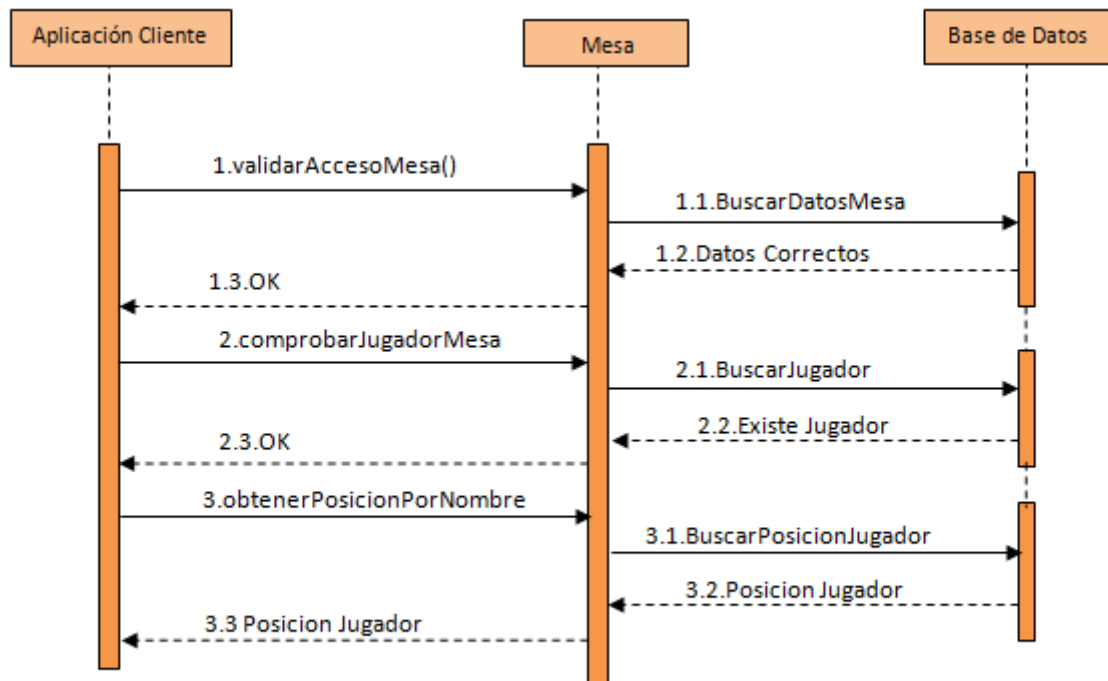


Figura 16 – Diagrama de secuencia de recuperar mesa (CU-05)

Como puede verse en este diagrama, cuando una aplicación cliente solicita la recuperación de una mesa se comprobará si los datos suministrados referentes a la mesa son correctos y si el usuario proporcionado estaba asociado a dicha mesa. Además, se le asignará al usuario la posición que tenía asignada a la mesa.

4.2.6 Jugar partida (CU-06)

En el diagrama que se muestra a continuación se pueden ver las interacciones de la aplicación cliente con las diferentes clases del servidor durante el transcurso de una partida de Cuatrola. Como se comentó al inicio de este capítulo, solo se mostrarán las principales interacciones y métodos necesarios para el desarrollo de una partida normal, ya que se necesitaría crear un diagrama de secuencia demasiado grande, lo que dificultaría en exceso la visualización y entendimiento del mismo.

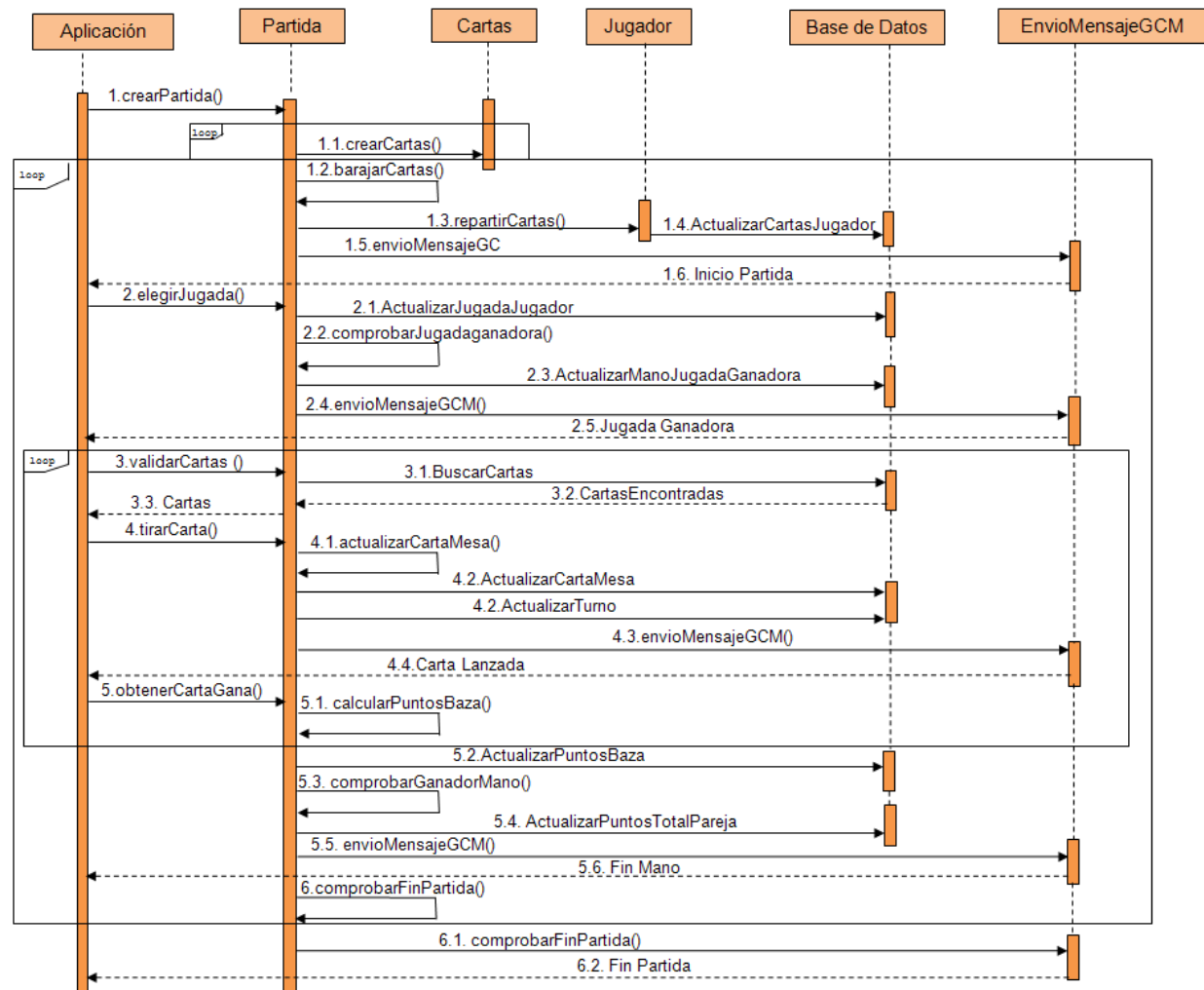


Figura 17 – Diagrama de secuencia de jugar partida (CU-06)

En el momento en que se asocia el cuarto usuario a la mesa, la aplicación cliente solicita al servidor comenzar la partida. Al inicio de una partida se generan las cartas, se barajan y se reparten a los cuatro jugadores de la partida. Una vez repartidas las cartas, cada una de las aplicaciones cliente elegirá el tipo de jugada, solicitando al servidor comprobar cuál es la más alta. Una vez hecho esta operación, el servidor enviará un mensaje a los diferentes clientes a través de la clase EnvíoMensajeGCM, indicando cual es la jugada mayor y el tipo de partida que se jugará. Además, le concederá el turno al jugador correspondiente, quién solicitará al servidor que valide cuales de las cartas puede tirar. El usuario lanzará la carta y la mesa la recogerá, y así hasta que cada usuario haya lanzado una carta. Tras esto, se comprobará que carta gana y que jugador la tiró, calculando los puntos ganados y sumándoselos a la pareja correspondiente. Este proceso se repetirá hasta que los diferentes usuarios hayan lanzado todas sus cartas, calculando los puntos de ambas parejas al final de cada mano e incrementando el marcador de partida de la pareja ganadora. Cuando una pareja alcance los 20 puntos la partida finalizará. Además, cuando un jugador realice una jugada y cuando se termine la partida el servidor deberá actualizar las estadísticas referentes a partidas y jugadas realizadas.

4.2.7 Enviar Estadísticas (CU-07)

En la Figura 18 se muestra el diagrama de secuencia relativo al caso de uso enviar estadísticas.

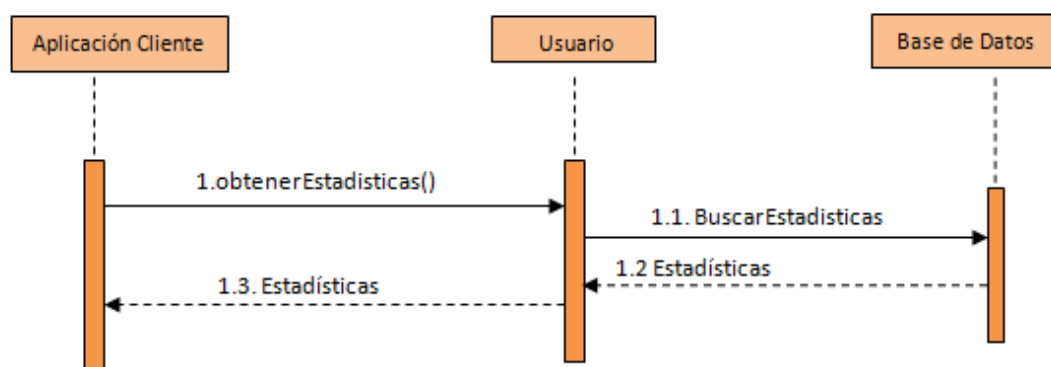


Figura 18 – Diagrama de secuencia de enviar estadísticas (CU-07)

Capítulo 5

Implementación del software

5.1 Diseño de Software

Análisis y el diseño del sistema a desarrollar, en este apartado se exponen las decisiones más importantes que se han tomado respecto a la implementación del servidor, tras haber realizado el diseño y el análisis del sistema.

5.1.1 Sincronización con GCM

Para poder sincronizar las cuatro aplicaciones clientes que participan en una partida de Cuatrola, después de estudiar diferentes posibilidades, se ha decidido utilizar GCM (Google Cloud Messaging). Como ya se ha explicado anteriormente como funciona este servicio de Google, en esta sección se explica el funcionamiento del servidor de la aplicación.

Para poder usar este servicio es necesario darse de alta como desarrollador en la plataforma de Google y crearse un proyecto. Cuando se crea un proyecto se le asigna un identificador de proyecto, el cual podrá ser utilizado por el servidor como identificador del mismo. Además, cuando un usuario se da de alta en el sistema, la aplicación cliente se registra en los servidores de Google obteniendo un identificador de registro. Este identificador es enviado al servidor, el cual lo almacena en la base de datos para posteriormente poder enviar mensajes a las aplicaciones cliente.

Durante el desarrollo de una partida de Cuatrola, cada vez que un usuario realiza alguna acción dentro de la partida o cuando éstos se están uniendo a la misma, el servidor debe enviar un mensaje específico a las cuatro aplicaciones cliente que participan en la partida. Este mensaje es enviado junto con el identificador de proyecto y los identificadores de registro de las aplicaciones cliente a los servidores de GCM, quien los reenviará a los diferentes dispositivos cliente.

Los mensajes que se envían son:

- UNIDO → para informar de la asociación de un usuario a una mesa.

- EMPEZAR → para indicar que ya están los cuatro usuarios unidos.
- CARTA_LANZADA → para informar que un usuario ha tirado una carta a la mesa.
- BAZA_FINALIZADA → para informar de que ha terminado una baza y debe comenzar una nueva.
- MANO_FINALIZADA → para informar que se ha terminado una mano y debe empezar una nueva.
- FIN_PARTIDA → para informar de que se alcanzado la puntuación necesaria para finalizar la partida.

5.1.2 Uso de JSON

Como se ha comentado en el punto anterior, el servidor necesitará enviar mensajes a GCM para que éste los reenvíe a las aplicaciones cliente.

Para el formato de estos mensajes se va utilizar JavaScript Object (JSON) [28]. Se trata de un formato estándar de texto plano para el intercambio de información. Una de las ventajas que tiene es que es independiente de cualquier lenguaje de programación, lo que permite el intercambio de datos entre componentes con diferentes tecnologías. Además, como la información a intercambiar serán datos simples, JSON es más sencillo de usar y más legible que XML [29].

5.1.3 Uso de DNS dinámico

Para que los usuarios puedan conectarse y usar la aplicación es necesario que el servidor sea accesible desde Internet. Además, sería apropiado poder configurar el equipo donde se monte éste con una dirección ip pública estática, ya que si la dirección ip es dinámica cada vez que ésta cambie del sería necesario actualizar la aplicación cliente con la nueva dirección del servidor.

Es por esto, por lo que se ha decidido usar un proveedor de DNS dinámico. Para la implementación del sistema se va usar el proveedor de DNS dinámico NO-IP [30], ya que se puede usar de manera gratuita y nos permite asignar un nombre de host a la dirección ip pública del servidor, haciendo que sea fácilmente accesible desde internet. Además, en caso de que la dirección ip sea dinámica, se encarga de comprobar si ésta cambia para asignar el nombre de host a la nueva dirección.

5.2 Resultados de las pruebas de aceptación

En este apartado se muestran los resultados obtenidos de las pruebas de aceptación definidas en el apartado diseño de las pruebas de aceptación (apartado 3.9).

Como se puede ver en la Tabla 11 todas las pruebas de aceptación definidas han sido superadas con éxito. Para presentar esta información se utilizará el formato definido en la plantilla 4 del Anexo 3.

Resultados pruebas de aceptación		
Id	Requisitos probados	Resultado
PA-01	RF-01	Superada
PA-02	RF-02	Superada
PA-03	RF-03	Superada
PA-04	RF-04	Superada
PA-05	RF-05, RF-08	Superada
PA-06	RF-05, RF-06, RF-07, RF-08	Superada
PA-07	RF-09, RF-10	Superada
PA-08	RF-11	Superada
PA-09	RF-12	Superada
PA-10	RF-13	Superada
PA-11	RF-14	Superada
PA-12	RF-15	Superada
PA-13	RF-16	Superada
PA-14	RF-17, RF18, RF-19	Superada
PA-15	RF-20, RF-21	Superada
PA-16	RF-20, RF-21, RF-22	Superada
PA-17	RF-22, RF-24, RF-25	Superada
PA-18	RF-26	Superada

Tabla 11 - Resultados de las pruebas de aceptación

Capítulo 6

Conclusiones y líneas futuras

6.1 Conclusiones sobre el proyecto

En esta sección se muestran las conclusiones alcanzadas una vez finalizado este proyecto fin de carrera analizando las diferentes dificultades encontradas durante todo su desarrollo junto con los resultados adquiridos.

6.1.1 Resultados obtenidos

Tras todo el proceso de construcción del sistema, se ha conseguido desarrollar una aplicación servidor que conectada con una aplicación cliente, permite jugar una partida al juego de cartas Cuatrola a cuatro usuarios conectado a través de internet desde sus Smartphones. Además, el sistema almacena y devuelve estadísticas personalizadas de cada uno de los usuarios guardados referentes a las partidas disputadas y jugadas realizadas ya sean ganadas o perdidas. Otro de los objetivos que se ha conseguido, es permitir jugar partidas de manera intermitente, es decir, empezar a jugar una partida, dejar la partida y continuarla más adelante.

Con todo lo descrito en el punto anterior se ha conseguido alcanzar cada uno de los objetivos propuestos al comienzo de este proyecto fin de carrera. También, con la construcción del sistema conjunto, aplicación cliente y servidor, se ha logrado crear una aplicación diferente para plataformas móviles Android, o al menos una aplicación difícil de encontrar. Navegando por la tienda de aplicaciones de Google para Android no se ha logrado hallar ninguna aplicación que permita disfrutar del juego de cartas Cuatrola de manera on-line e incluso, según se ha podido comprobar en los comentarios de los usuarios en la tienda de aplicaciones, existe un potencial interés en una aplicación de este tipo.

Además, se ha conseguido que la aplicación servidor pueda comunicarse fácilmente con las diferentes aplicaciones cliente a través de internet. También se ha logrado que el servidor permita que se puedan jugar varias partidas distintas a la vez.

6.1.2 Dificultades del proyecto

La mayor dificultad que hubo que vencer fue a la hora de programar las diferentes funcionalidades del componente encargado de gestionar la partida. El problema en cuestión, fue porque en un principio se creía que cada vez que una aplicación cliente solicitaba al servidor ejecutar un determinado fichero con código PHP, éste continuaría la ejecución por donde lo dejó la anterior aplicación cliente, pero cada vez que se ejecuta el código de un fichero se realiza desde el inicio, por lo que hubo que cambiar de mentalidad y adaptarse a una manera de programar a la que no se estaba acostumbrado.

Una de las dificultades mas grandes afrontadas durante la realización de este proyecto fin de carrera vino a la hora de sincronizar las cuatro aplicaciones cliente que participan en una misma partida, ya que cada vez que una de ellas realiza una operación en el sistema es necesario informar a todas sobre el resultado de la operación realizada. Después de realizar un estudio durante la fase de análisis de las diferentes posibilidades a utilizar, se decidió usar los servicios que proporciona Google Cloud Messaging. Así, cuando una aplicación cliente realiza una operación sobre el sistema, el servidor envía un mensaje informativo, diferente según la operación realizada, a los servidores de Google, para que estos los reenvíe a las diferentes aplicaciones cliente sin que ni si quiera éstas estén esperando una respuesta.

Otra de los problemas encontrados vino cuando se intentaron utilizar los servicios de Google Cloud Messaging por primera vez. La mayor dificultad para usar estos servicios surgió a la hora de registrarse en la plataforma de desarrolladores de Google, ya que además de registrarse, es necesario crearse un proyecto y asociarle dichos servicios, y aunque parezca algo sencillo, no lo es, toda la documentación que se encontró estaba desactualizada u omitía pasos.

6.1.3 Conclusiones personales

El mayor reto de este proyecto ha sido el desconocimiento total de las tecnologías a usar y el cómo programar y configurar una aplicación servidor, lo que ha supuesto una motivación más para la realización de este proyecto.

A pesar de todas las dificultades encontradas, al final se han conseguido alcanzar los objetivos propuestos. Para mí todo este proceso ha sido muy motivador y provechoso, ya que además de conseguir una meta académica y un reto personal, he podido conocer y aprender diferentes tecnologías, herramientas y un nuevo lenguaje de programación como PHP. Gracias a la realización de este proyecto he logrado entender cómo funciona una aplicación cliente/servidor y saber cómo poder comunicarte y sincronizarte con una aplicación cliente para Smartphones con Android.

Además, durante la realización del proyecto he tenido que realizar tareas en equipo con mi compañero, y he de decir que ha sido una experiencia bastante satisfactoria y enriquecedora, tanto para lo profesional como lo personal.

6.2 Líneas futuras

En esta sección se detallan las posibles mejoras y extensiones que se pueden realizar sobre la aplicación servidor desarrollada a lo largo de este proyecto. Tales mejoras y extensiones se han ido identificando a lo largo de todo el proceso de construcción de este proyecto.

6.2.1 Parametrización de los máxima puntuación para finalizar una partida

Una de las opciones que podría añadirse es la parametrización de los puntos totales con los que finalizaría una partida de Cuatrola. El desarrollo del proyecto se ha realizado pensando en esta extensión, y la base de datos contiene un campo donde se indica el número de puntos a conseguir para finalizar el juego. Se podría permitir al usuario elegir dicha puntuación cuando crea una mesa y actualizar dicho valor en la base de datos. En este proyecto se ha utilizado por defecto una puntuación final de veinte puntos, que suele ser la forma típica de jugar.

6.2.2 Creación de mesas públicas

Otra de las opciones a añadir sería dar la posibilidad de crear mesas públicas, es decir, mesas sin contraseña de acceso y que cualquier usuario pueda entrar a jugar.

Para ello sería necesario obtener un listado de todas las mesas públicas creadas pendientes de llenarse y enviárselo a la aplicación cliente para que el usuario pueda ver y elegir la mesa que quiera. Con lo que en el servidor tendría que añadir una consulta a la base de datos y obtener las mesas públicas creadas y que no están llenas y enviar dicha información a la aplicación cliente cuando lo solicite.

6.2.3 Creación de chat para la partida

Otra posible extensión sería la creación de un chat dentro de cada una de las partidas para que los usuarios puedan hablar y realizar comentarios sobre la partida que están disputando.

6.2.4 Aumentar las estadísticas proporcionadas.

También, se podría mejorar la información mostrada en las estadísticas ofreciendo mayor información. Se podría incluir el número de partidas ganadas y perdidas con respecto a las jugadas, el tanto por ciento de victorias y derrotas y lo mismo para cada una de las posibles jugadas. Incluso se podrían añadir estadísticas de las partidas y jugadas realizadas con las diferentes parejas con las que se ha jugado al juego.

6.2.5 Introducción de un ranking de puntuaciones.

Otra extensión de la aplicación sería crear un ranking de puntuaciones, es decir, crear un marcador global donde se muestren los jugadores que más partidas han ganado y que más jugadas diferentes han realizado con éxito.

6.2.6 Introducción de diferentes juegos de cartas.

Otra de las posibles mejoras a realizar sería incluir en la aplicación diferentes juegos de cartas, dando la posibilidad al usuario de elegir a que juego quiere jugar. Esta extensión llevaría bastante tiempo, ya que para cada uno de los juegos a introducir sería necesario implementar el algoritmo, sin embargo con unas pequeñas modificaciones en el código de la Cuatrola se podría adaptar para jugar al Tute o la Brisca.

6.2.7 Creación de una aplicación web para jugar a Cuatrola

Otra mejora detectada sería la opción de poder jugar al juego no solo en dispositivos móviles, sino también en un navegador web a través de cualquier dispositivo que lo incluya. Como el código del servidor se ha desarrollado en PHP, sería fácil adaptarlo e incrustarlo en código HTML, lo único que habría que realizar sería crear una interfaz gráfica.

Referencias

[1] Internet Trends 2015 – Code Conference [En línea]

Disponible: <http://www.kpcb.com/internet-trends>, último acceso Octubre 2015.

[2] Estadísticas sobre aplicaciones para Smartphones [En línea]

Disponible: <http://www.statista.com>, último acceso Agosto 2015.

[3] Ingresos y crecimiento de los juegos en dispositivos móviles [En línea]

Disponible: <http://www.emarketer.com/Article/Mobile-Game-Revenues-Grow-165-2015-Surpassing-3-Billion/1012063?ecid=PR1007>, último acceso Octubre 2015.

[3] Ingresos y crecimiento de los juegos en dispositivos móviles [En línea]

Disponible: <http://www.emarketer.com/Article/Mobile-Game-Revenues-Grow-165-2015-Surpassing-3-Billion/1012063?ecid=PR1007>, último acceso Octubre 2015.

[4] Apple App Store desde iTunes [En línea]

Disponible: <http://www.apple.com/itunes>, último acceso Agosto 2015.

[5] Google Play Store [En línea]

Disponible: <https://play.google.com/store>, último acceso Septiembre 2015.

[6] Amazon App Store [En línea]

Disponible: <http://www.amazon.com/mobile-apps/b?node=2350149011>, último acceso Agosto 2015.

[7] Blackberry World App Store [En línea]

Disponible: <https://appworld.blackberry.com/webstore/>, último acceso Agosto 2015.

[8] Microsoft Windows Phone App Store [En línea]

Disponible: <https://www.microsoft.com/en-us/store/apps/windows-phone>, último acceso Agosto 2015.

[9] Mysql [En línea]

Disponible: <https://www.mysql.com>, último acceso Agosto 2015.

[10] Oracle [En línea]

Disponible: <https://www.oracle.com/database/index.html>, último acceso Agosto 2015.

[11] Sql Server [En línea]

Disponible: <http://www.microsoft.com/es-es/server-cloud/products/sql-server/default.aspx>,
último acceso Agosto 2015.

[12] María Pérez Marqués, *MYSQL. Diseño, Programación y Administración de Bases de Datos*, Editorial Createspace, 2013.

[13] JAVA [En línea]

Disponible: <https://www.java.com/es>, último acceso Agosto 2015.

[14] PHP [En línea]

Disponible: <https://secure.php.net/manual/es/index.php>, último acceso Septiembre 2015

[15] PERL [En línea]

Disponible: <https://www.perl.org>, último acceso Septiembre 2015.

[16] Python [En línea]

Disponible: <https://www.python.org>, último acceso Septiembre 2015.

[17] Apache HTTP Server [En línea]

Disponible: <https://httpd.apache.org>, último acceso Septiembre 2015.

[18] Juan Carlos Martínez de Llardua, *Administración de Oracle 11g*, Editorial Sintesis, 2012.

[19] Santiago Media Serrano, *SQL Server 2104: Soluciones prácticas de administración*, Editorial RA-MA, 2015.

[20] JSP [En línea]

Disponible: <http://www.oracle.com/technetwork/java/javase/jsp/index.html>, último acceso Septiembre 2015.

[21] Javier Urbaneja Fan, *JSP*, Editorial Anaya Multimedia, 2008.

[22] Randal L. Schwartz, *Programming Perl*, Editorial O Reilly & Associates, 2000.

[23] Luke Weeling y Laura Thomson, *Desarrollo web con PHP y MySQL*, Editorial Anaya Multimedia, 2009.

[24] IIS [En línea]

Disponible: <https://www.iis.net/>, último acceso Septiembre 2015.

[25] Mohamed J. Kabir, *La biblia del servidor Apache 2*, Editorial Anaya Multimedia, 2003.

[26] XAMPP [En línea]

Disponible: <https://www.apachefriends.org/es/index.html>, último acceso Octubre 2015.

[27] Google Cloud Messaging [En línea]

Disponible: <https://developers.google.com/cloud-messaging/>, último acceso Julio 2015.

[28] JSON [En línea]

Disponible: <http://www.json.org>, último acceso Septiembre 2015.

[29] XML [En línea]

Disponible: <http://www.w3.org/XML>, último acceso Septiembre 2015.

[30] NO-IP [En línea]

Disponible: <http://www.noip.com>, último acceso Octubre 2015.

[31] Windows 7 [En línea]

Disponible: <http://windows.microsoft.com/en-us/windows/windows-help#windows=windows-7>, último acceso Junio 2015

[32] Visual Paradigm [En línea]

Disponible: <http://www.visual-paradigm.com/>, último acceso Junio 2015

[33] Notepad ++ [En línea]

Disponible: <https://notepad-plus-plus.org/>, último acceso Mayo 2015

[34] Microsoft Office 2013 [En línea]

Disponible: <https://products.office.com/en-us/compare-microsoft-office-products>, último acceso Junio 2015

[35] Microsoft Project 2013 [En línea]

Disponible:

http://www.microsoftstore.com/store/mseea/es_ES/list/Project/categoryID.66235900, último
acceso Junio 2015

[36] Android OS 5.1.1 Lollipop [En línea]

Disponible: https://www.android.com/intl/es_es/, último acceso Junio 2015

Acrónimos

KPCB	<i>Kleiner Perkins Caufield & Byers</i>
PHP	<i>Hypertext Pre-Processor</i>
PERL	<i>Practical Extraction and Report Language</i>
JSP	<i>JavaServer Pages</i>
HTTP	<i>Hypertext Transfer Protocol</i>
HTTPS	<i>Hypertext Transfer Protocol Secure</i>
IIS	<i>Internet Information Services</i>
FTP	<i>File Transfer Protocol</i>
SMTP	<i>Simple Mail Transfer Protocol</i>
NNTP	<i>Network News Transfer Protocol</i>
XAMPP	<i>X Apache MySQL PHP PERL</i>
GCM	<i>Google Cloud Messaging</i>
JSON	<i>JavaScript Object Notation</i>
XML	<i>Extensible Markup Language</i>
IP	<i>Internet Protocol</i>

DNS

Domain Name System

Anexo 1: Gestión del proyecto

En este anexo se detallan los diferentes aspectos concernientes a la gestión del proyecto que incluyen la planificación del trabajo, los medios técnicos utilizados para el desarrollo de todo el proyecto así como el análisis económico del mismo.

1. Planificación del trabajo

En este apartado se detallan la planificación inicial del proyecto y el desarrollo real del mismo, así como un análisis de las desviaciones encontradas entre la planificación inicial y el tiempo real dedicado al desarrollo completo de la aplicación

1.1 Planificación inicial

En esta sección se detalla la planificación inicial planteada para el desarrollo de este proyecto fin de carrera. En ella se pueden ver las diferentes etapas en las que se ha dividido el proyecto junto

con la estimación del esfuerzo a realizar para cada una de las tareas identificadas. En la elaboración de esta planificación se han tenido en cuenta jornadas de 40 horas semanales divididas en 5 jornadas diarias de 8 horas cada una.

La planificación inicial del proyecto abarca desde el 23 de abril hasta el 15 de septiembre de 2015, lo que hace un total de 104 días laborables lo que equivale a 832 horas de trabajo.

En la Tabla 12 se puede ver el detalle de la planificación inicial del proyecto con el desglose de las diferentes tareas a realizar.

Planificación inicial del Proyecto Fin de Carrera			
Tarea	Días	Inicio	Fin
Planificación inicial	2	23/04/2015	24/04/2015
Estudio del estado del arte	4	27/04/2015	30/04/2015
Análisis de aplicaciones existentes	2	27/04/2015	28/04/2015
Aplicaciones actuales	2	29/04/2015	30/04/2015
Análisis	12	01/05/2015	18/05/2015
Arquitectura del sistema	5	01/05/2015	07/05/2015
Estudio tecnológico	4	08/05/2015	13/05/2015
Definición de casos de uso	1	14/05/2015	14/05/2015
Definición de requisitos software	2	15/05/2015	18/05/2015
Diseño	16	19/05/2015	09/06/2015
Diseño de software	8	19/05/2015	28/05/2015
Diagramas de secuencia	8	29/05/2015	09/06/2015
Implementación	40	10/06/2015	04/08/2015
Pruebas	4	05/08/2015	10/08/2015
Documentación	26	11/08/2015	15/09/2015
TOTAL	104	23/04/2015	15/09/2015

Tabla 12 – Planificación inicial del proyecto detallada

Como se puede observar en esta planificación la mayor parte del tiempo estimado para el desarrollo del proyecto está dirigido a la implementación del mismo. Además, para las tareas de documentación y diseño también es necesario invertir gran cantidad de tiempo.

En la Figura 19 se muestra el diagrama de Gantt con la planificación inicial del proyecto y todas las tareas desglosadas.

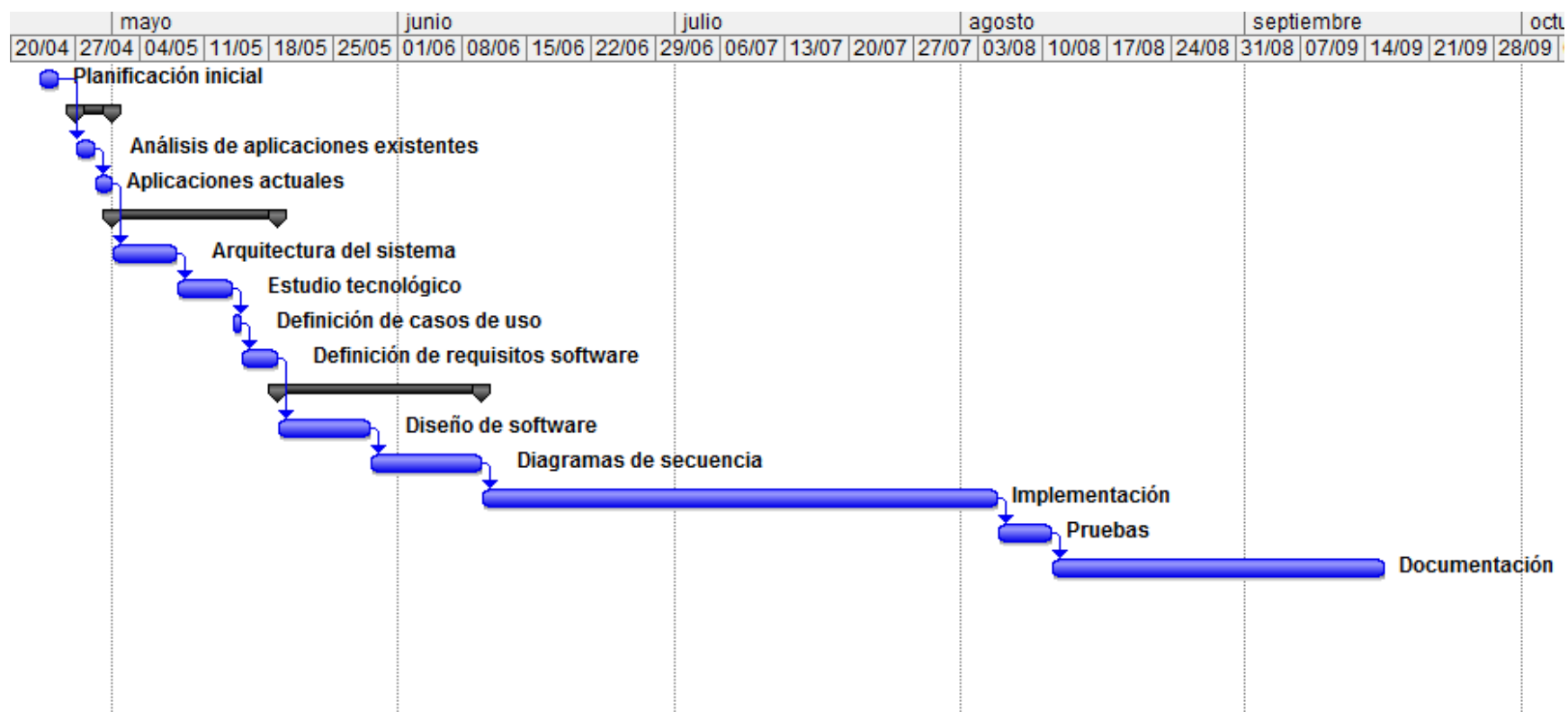


Figura 19 - Planificación inicial del proyecto

1.2 Desarrollo real del proyecto

En este apartado se detalla el desarrollo real del proyecto y se equipara con la planificación inicial del mismo para analizar las diferentes desviaciones sufridas a lo largo de la elaboración de todo el proyecto.

En la siguiente tabla se muestra el desarrollo real del proyecto y del desglose de cada una de las tareas realizadas.

Planificación inicial del Proyecto Fin de Carrera			
Tarea	Días	Inicio	Fin
Planificación inicial	2	23/04/2015	24/04/2015
Estudio del estado del arte	4	27/04/2015	30/04/2015
Análisis de aplicaciones existentes	2	27/04/2015	28/04/2015
Aplicaciones actuales	2	29/04/2015	30/04/2015
Análisis	14	01/05/2015	20/05/2015
Arquitectura del sistema	5	01/05/2015	07/05/2015
Estudio tecnológico	5	08/05/2015	14/05/2015
Definición de casos de uso	2	15/05/2015	18/05/2015
Definición de requisitos software	2	19/05/2015	20/05/2015
Diseño	18	21/05/2015	15/06/2015
Diseño de software	9	21/05/2015	02/06/2015
Diagramas de secuencia	9	03/06/2015	15/06/2015
Implementación	43	16/06/2015	13/08/2015
Pruebas	5	14/08/2015	20/08/2015
Documentación	30	21/08/2015	01/10/2015
TOTAL	116	23/04/2015	01/10/2015

Tabla 13 – Desarrollo real del proyecto detallado

La Figura 20 muestra el diagrama de Gantt del desarrollo real del proyecto. En ella se puede observar que el reparto de tiempo dedicado es bastante parecido a la estimación realizada inicialmente, aunque como suele pasar en la mayoría de proyectos la duración final del mismo ha

sido superior a la estimada. La duración final del proyecto ha sido de 116 días de trabajo, lo que equivale a 928 horas de dedicación.

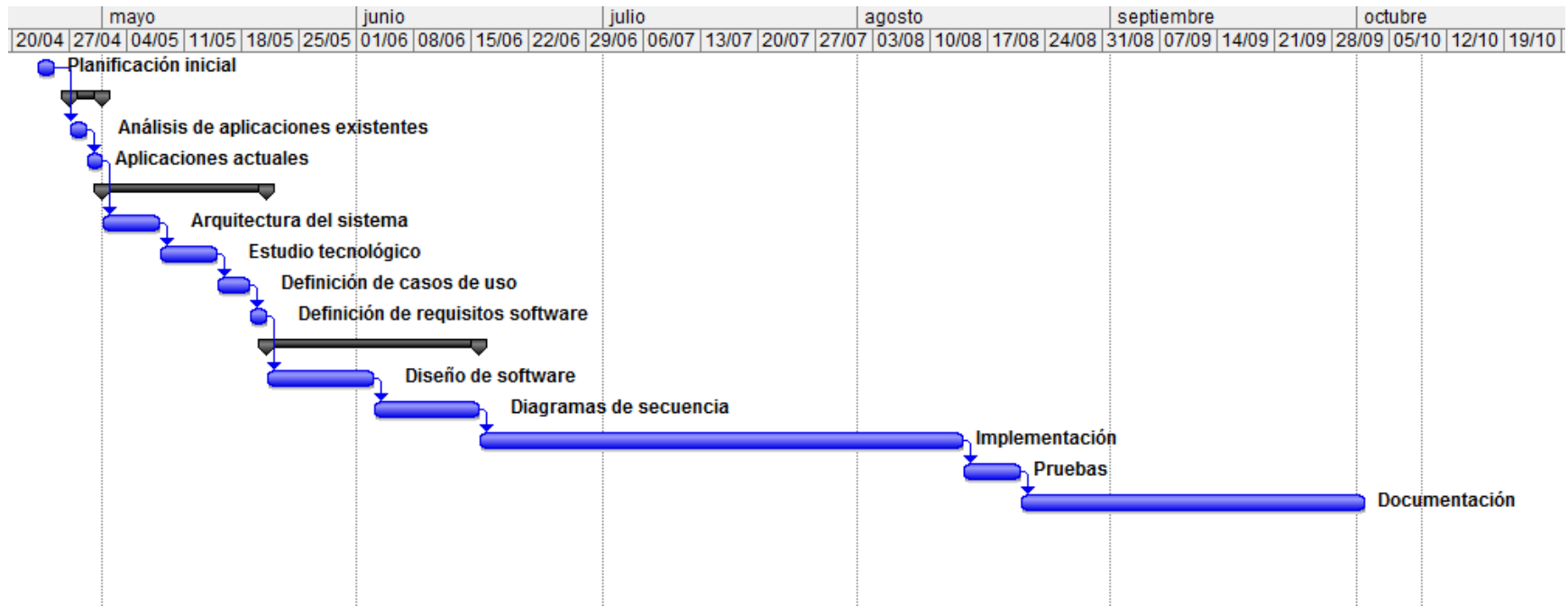


Figura 20 - Desarrollo real del proyecto

En la tabla 14 se muestra la comparativa entre la planificación inicial estimada y el desarrollo real del proyecto y el desglose de cada una de las tareas. En esta tabla se puede ver de manera rápida e intuitiva las desviaciones ocurridas durante el desarrollo del proyecto.

Comparativa entre la planificación estimada y el desarrollo real				
Tarea	Planificado (días)	Real (días)	Diferencia (días)	Variación (%)
Planificación inicial	2	2	0	0,0%
Estudio del estado del arte	4	4	0	0,0%
Análisis de aplicaciones existentes	2	2	0	0,0%
Aplicaciones actuales	2	2	0	0,0%
Análisis	12	14	2	16,7%
Arquitectura del sistema	5	5	0	0,0%
Estudio tecnológico	4	5	1	25,0%
Definición de casos de uso	1	2	1	100,0%
Definición de requisitos software	2	2	0	0,0%
Diseño	16	18	2	12,5%
Diseño de software	8	9	1	12,5%
Diagramas de secuencia	8	9	1	12,5%
Implementación	40	43	3	7,5%
Pruebas	4	5	1	25,0%
Documentación	26	30	4	15,4%
TOTAL	104	116	12	11,5%

Tabla 14 – Comparativa entre planificación estimada y desarrollo real del proyecto

Como puede verse en la tabla anterior, el proyecto se ha alargado 12 días más de lo esperado y la desviación total del mismo ha sido del 11,5%. Esta desviación se ha debido principalmente a la subestimación de las tareas de análisis, diseño, implementación, pruebas y sobre todo documentación. Esta subestimación ha sido originada por los diversos problemas que han surgido a lo largo del desarrollo y que no se habían tenido en cuenta en un principio. A pesar de todo esto, la desviación no ha sido muy grande.

Los principales problemas encontrados supusieron realizar un nuevo análisis y cambiar algunos diseños, lo que impactó en las tareas de implementación, pruebas y documentación.

2. Medios técnicos empleados para el proyecto

En esta sección se describen los medios que se ha utilizado a lo largo de todo el proceso de desarrollo del proyecto.

La siguiente tabla muestra las distintas herramientas utilizadas junto con una descripción de cada una de ellas.

Herramienta	Descripción
Windows 7 [31]	Sistema operativo sobre el que se han utilizado el resto de herramientas para el desarrollo de este proyecto fin de carrera.
Visual Paradigm [32]	Herramienta de modelado UML utilizada para la creación de todos los diagramas utilizados en la fase de análisis y diseño.
Notepad ++ [33]	Herramienta de edición de texto utilizada para el desarrollo.
Microsoft Office 2013 (Windows) [34]	Suite ofimática utilizada para la documentación.
Microsoft Project 2013 (Windows) [35]	Aplicación para la gestión de planificaciones utilizada para realizar la planificación y el seguimiento del proyecto.
Android OS 5.1.1 [36]	Sistema operativo utilizado para las pruebas en un Smartphone.
XAMPP	Es una distribución de Apache que contiene MySQL, PHP y PERL. Permite configurar e instalar fácilmente un servidor.
NO-IP	Herramienta para asignar un nombre de host a la ip pública del servidor

Tabla 15 – Herramientas utilizadas en el proyecto

Además del software utilizado, se ha necesitado usar una serie de equipos y dispositivos para el desarrollo del proyecto. En la Tabla 16 se especifican cada uno de los equipos y dispositivos utilizados.

Equipo	Descripción
Portátil: Asus RJ510JF Intel® Core™ i7 8GB RAM	Ordenador con el que se ha desarrollado todo el proyecto fin de carrera.
Impresora: HP envy 4500	Utilizada para la impresión de los documentos necesarios para la realización del proyecto.
Smartphone: Honor 6	Utilizado para realizar las pruebas de la aplicación desarrollada en el proyecto.

Tabla 16 – Equipos utilizados en el proyecto

3. Análisis económico del proyecto

En este apartado se expone el análisis económico del proyecto que incluye la metodología usada para la estimación de los costes, el presupuesto inicial, el presupuesto para el cliente y el presupuesto real del proyecto realizado.

3.1 Metodología de estimación de costes

La estimación de los costes para el desarrollo del proyecto se ha realizado teniendo en cuenta costes indirectos y directos.

Los costes directos son aquellos que están relacionados directamente con los gastos del desarrollo del proyecto. Entre estos conceptos se incluyen los gastos de mano de obra, equipos y dispositivos utilizados, herramientas de software necesitadas, dietas y gastos de movilidad.

Los costes directos se corresponden con los conceptos que están relacionados directamente con el desarrollo del proyecto. Entre estos conceptos se encuentran los gastos de mano de obra, equipos, herramientas de software, dietas y gastos de viaje.

Los gastos relacionados con la mano de obra se estimarán teniendo en cuenta que se está empleado directamente por una empresa. Los precios de los equipos y dispositivos se valorarán según el precio oficial de venta a particulares, el valor de las dietas se fija como pactadas entre empleado y empresa y los gastos de movilidad se establecen en función del kilometraje del desplazamiento.

Los costes indirectos son aquellos costes que se necesitan para el desarrollo del proyecto pero que no están relacionados directamente con el desarrollo del sistema objetivo. Entre estos gastos se encuentran los gastos de teléfono, la conexión a internet y el alquiler del inmueble para la realización de las diferentes actividades.

Entre estos gastos se encuentra el teléfono, la conexión a internet y el alquiler del inmueble en el que se realiza la actividad, entre otros. Para el desarrollo del proyecto se ha estimado que los costes indirectos suponen un 20% de los costes directos.

3.2 Presupuesto inicial

En este apartado se muestra el proyecto inicial estimado donde se especificarán los gastos presupuestados para todo el proyecto.

El 21% de IVA no está incluido en ninguno de los gastos detallados en este apartado, solo se añadirá en el presupuesto total del proyecto.

3.2.1 Gastos de personal

La Tabla 17 especifica los gastos de personal. Para su cálculo se ha tomado como base la utilización de un único desarrollador con perfil junior, ya que jamás he trabajado como desarrollador y no conocía ninguna de las tecnologías a usar. Se han dedicado 8 horas diarias a lo largo de todo el proyecto, durante 104 días laborales, que es la duración que se estimó inicialmente para la realización del proyecto. El coste hombre mes del recurso se ha tomado analizando los sueldos en diferentes ofertas de trabajo del mercado actual. Este valor es un importe bruto al que se ha habido que añadir el importe a pagar a la Seguridad Social, 23,6% para el régimen general de la Seguridad Social.

Recurso	Dedicación	Coste hombre/mes	Coste bruto	Seg. Social	Coste neto
Desarrollador Junior	4,95 hombres/mes	1.150 €	5.692,50 €	1.343,43 €	7.035,93 €

Tabla 17 - Gastos de personal

Como se puede ver en la tabla, los cálculos estiman una dedicación de 4,95 hombres mes, lo que implica un coste total bruto de 5.692,5 € a los que hay que sumar el coste de la Seguridad Social, resultando un coste total de personal de 7.035,93 €.

3.2.2 Gastos de equipos

En este apartado se detallan los gastos referentes con los diferentes equipos utilizados a lo largo de la realización del proyecto. Los equipos que se han usado son un ordenador portátil, una impresora y un Smartphone. En la Tabla 18 se detallan los costes imputables para cada uno de los equipos y los datos utilizados para calcular dichos costes: porcentaje de uso, coste y periodo de depreciación.

El periodo de depreciación para el ordenador y la impresora es el utilizado en el ámbito de la Administración General mientras que el del teléfono es el periodo de lanzamiento entre nuevos modelos más avanzados. Para el desarrollo de este proyecto se ha estimado que se realizará en 4,95 meses según la planificación inicial.

Equipo	Costes sin iva	Dedicación	Periodo depreciación	Coste imputable
Portátil:	629,75 €	3,47 meses	36 meses	60,70 €
Asus RJ510JF Intel® Core™ i7 8GB RAM				
Impresora:	41,83 €	3,47 meses	36 meses	4,03 €
HP envy 4500				
Smartphone:	237 €	3,47 meses	24 meses	34,27 €
Honor 6				
			Total imputable	99,00 €

Tabla 18 - Gastos de equipos

3.2.3 Gastos de software

En este apartado se especifican los gastos relacionados con el software utilizado para la realización del proyecto. En la Tabla 19 se puede ver el detalle de los costes imputables para cada una de las aplicaciones usadas de la misma forma que se hizo en la sección anterior con los gastos de equipos.

En esta tabla solo muestra el software comentado en la sección 2 de este anexo que no es gratuito.

Software	Costes sin iva	Dedicación	Periodo depreciación	Coste imputable
Windows 7	111,57 €	3,47 meses	36 meses	10,75 €
Visual Paradigm	537,19 €	3,47 meses	36 meses	51,78 €
Microsoft Office 2013	123,14 €	3,47 meses	36 meses	11,87 €
Microsoft Project 2013	635,53 €	3,47 meses	36 meses	61,26 €
			Total imputable	135,66 €

Tabla 19 - Gastos de software

3.2.4 Gastos de consumibles

En este apartado se detallan los gastos de consumibles necesitados a lo largo de todo el proyecto. Estos gastos constan de los costes dedicados cartuchos de tinta para la impresora y material de oficina.

La Tabla 20 muestra el resumen de los gastos generados por los consumibles utilizados en la realización del proyecto.

Consumible	Coste unitario sin iva	Cantidad	Coste total consumible
Cartuchos de impresora	64,75 €	1	64,75 €
Material oficina	18,50 €	1	18,50 €
		Coste total	83,25 €

Tabla 20 - Gastos de consumibles

3.2.5 Gastos de viajes y dietas

En este apartado se describen los gastos de movilidad y dietas necesitadas a lo largo de todo el proyecto.

Se considera al tutor de este proyecto como si fuese el cliente final y por tanto, como el cliente está localizado en Leganés, fuera del término municipal de la ciudad de Madrid, los gastos de viaje se han de imputar al proyecto y se calcularán en base a los kilómetros diarios realizados con un coste de 0,12 €/Km y teniendo en cuenta que la distancia media recorrida por desplazamiento es de 78 Km. Además, se ha asignado un importe diario en concepto de dietas de 9,5 € al día.

La Tabla 21 muestra los gastos en concepto de viajes y dietas.

Descripción gasto	Coste unitario sin iva	Cantidad	Coste total concepto
Gastos Viajes Leganés	0,12 €/Km	468 Km	56,16 €
Dietas	9,5€/día	104	988,00 €
Coste total			1.044,16 €

Tabla 21 - Gastos de viajes y dietas

3.2.6 Costes directos

En este apartado se detallan los costes directos asociados a la realización del proyecto que son el total de los diferentes conceptos calculados en las secciones anteriores: gastos de personal, gastos de equipos, gastos de software, gastos de consumibles y gastos de desplazamiento y dietas.

En la Tabla 22 se puede ver la suma de estos conceptos dando como resultado el total de costes directos del proyecto.

Concepto	Coste
Gastos de personal	7.035,93 €
Gastos de equipo	99,00 €
Gastos de software	135,66 €
Gastos de consumibles	83,25 €
Gastos de viajes y dietas	1.044,16 €
Costes directos	8.398,00 €

Tabla 22 - Costes directos

3.2.7 Costes indirectos

Los costes indirectos se han calculado de acuerdo a lo visto en la sección 3.1 de este anexo como el 20% de los costes directos.

De este modo y con los resultados obtenidos en el apartado anterior, los costes indirectos del proyecto ascienden a 1694,54 €.

3.2.8 Estimación de costes

En este apartado se detalla la estimación de los costes del proyecto a partir de los cálculos realizados en las secciones anteriores.

La Tabla 23 muestra los costes identificados y la suma de todos ellos sin IVA, añadiéndoselo posteriormente para obtener así la estimación inicial.

Concepto	Coste
Gastos de personal	7.035,93 €
Gastos de equipo	99,00 €
Gastos de software	135,66 €
Gastos de consumibles	83,25 €
Gastos de viajes y dietas	1.044,16 €
Costes directos	8.398,00 €
Costes indirectos	1.679,60 €
TOTAL sin IVA	10.077,60 €
IVA (21%)	2.116,30 €
TOTAL con IVA	12.193,90 €

Tabla 23 – Estimación de costes

3.3 Presupuesto para el cliente

En este apartado se detalla el presupuesto a presentar al cliente. En este presupuesto se agrupan todos los gastos vistos anteriormente y se añade un porcentaje de riesgo para poder cubrir los diferentes gastos de los imprevistos aparecidos. Además, se incluyen los beneficios esperados por el desarrollo de este proyecto. El porcentaje de riesgos definido para el proyecto es del 12%, dicho valor se ha definido de acuerdo a los valores que se han visto en los diferentes proyectos analizados.

También, se ha definido un porcentaje del 15% en concepto de beneficios. Este nuevo concepto se incluirá en el presupuesto. Se ha decidido utilizar un 15% porque entra dentro del rango de valores vistos en otros proyectos analizados.

Concepto	Coste
Gastos de personal	7.035,93 €
Gastos de equipo	99,00 €
Gastos de software	135,66 €
Gastos de consumibles	83,25 €
Gastos de viajes y dietas	1.044,16 €
Costes directos	8.398,00 €
Costes indirectos	1.679,60 €
TOTAL gastos sin riesgo	10.077,60 €
Riesgo (12%)	1.209,31 €
TOTAL gastos sin beneficios	11.286,91 €
Beneficios (15%)	1.693,04 €
TOTAL gastos sin iva	12.979,95 €
IVA (21%)	2.725,79 €
TOTAL con IVA	15.705,74 €

Tabla 24 - Presupuesto para el cliente

3.4 Coste final y análisis de la desviación

En esta sección se detalla el coste final que ha supuesto el proyecto y se compara con el coste estimado inicialmente. Debido a la desviación en la planificación inicial del

proyecto, se puede esperar que el presupuesto final también sufra una desviación con respecto al estimado.

La Tabla 25 muestra una comparativa entre el presupuesto inicial estimado y el coste final del proyecto, indicando la variación en cada uno de los conceptos y en el total. Para el cálculo del coste final del proyecto se han tomado los mismos costes base modificando los meses reales de desarrollo del proyecto así como el número de horas hombre para calcular así el coste real del proyecto.

Concepto	Coste presupuestado	Coste real	Desviación
Gastos de personal	7.035,93 €	7.479,88 €	443,95 €
Gastos de equipo	99,00 €	110,41 €	11,41 €
Gastos de software	135,66 €	138,06 €	2,40 €
Gastos de consumibles	83,25 €	83,25 €	0,00 €
Gastos de viajes y dietas	1.044,16 €	1.158,16 €	114,00 €
Costes directos	8.398,00 €	8.969,76 €	571,76 €
Costes indirectos	1.679,60 €	1.793,95 €	114,35 €
TOTAL	10.077,60 €	10.763,71 €	686,11 €

Tabla 25 - Coste real del proyecto comparado con el presupuestado

Como se puede observar en la tabla anterior existe una desviación en el coste final de 686,11 €. Esta variación entre el coste presupuestado y el coste final no puede afectar en el precio a facturar al cliente, por lo que se tendrá que descontar del beneficio presupuestado que como se puede apreciar en la tabla 24 es de 1.693,04 €. Por lo que si restamos a estos beneficios la variación en los costes totales, arroja una ganancia de 1.006,93 €. Sumando a estas ganancias el importe íntegro reservado para riesgos se consigue un beneficio de: 2.180,34 €.

Como consecuencia de las desviaciones sobre la planificación inicial cabía esperar estos resultados ya que esta variación afecta no solo a un incremento de gastos de personal sino que también afecta prácticamente el resto de gastos directos e indirectos, aun así se ha conseguido un beneficio aceptable.

Anexo 2: Manual de instalación y configuración del servidor

1. Introducción

Esta aplicación se ha desarrollado para que los usuarios puedan jugar al juego de cartas Cuatrola con gente real y de manera online. La aplicación se divide en dos componentes principalmente: aplicación cliente y aplicación servidor.

Este manual pretende explicar cómo instalar y configurar la aplicación servidor para poder ofrecer los servicios solicitados a la aplicación cliente.

2. Requisitos previos e instalación

Para poder instalar el software necesario para la aplicación servidor, solo es necesario disponer de un ordenador con sistema operativo Windows, OS X o Linux y con conexión a Internet. Además, el ordenador debe disponer al menos de 512 MB de RAM y 80 MB disponibles en el disco duro.

3. Instalación y configuración

En esta sección se explica la instalación del software necesario para el servidor, así como su configuración.

3.1 Instalación de XAMPP

Lo primero que hay que instalar es el servidor web HTTP Apache, en concreto la distribución XAMPP. Esta distribución puede instalarse en la mayoría de sistemas operativos y además del servidor Apache, también trae consigo MySQL y el intérprete de PHP, ambos necesarios para poder ofrecer los servicios a la aplicación cliente.

XAMPP, se puede descargar de manera gratuita de su página oficial: <https://www.apachefriends.org/es/download.html>.

Una vez descargado, hay que proceder a la instalación del mismo. Cuando se ejecuta el instalador, en caso de tener un antivirus instalado, aparece una ventana de aviso indicando que éste puede interferir sobre la instalación, es necesario aceptar este mensaje para proseguir. En la siguiente figura se muestra el mensaje de aviso:

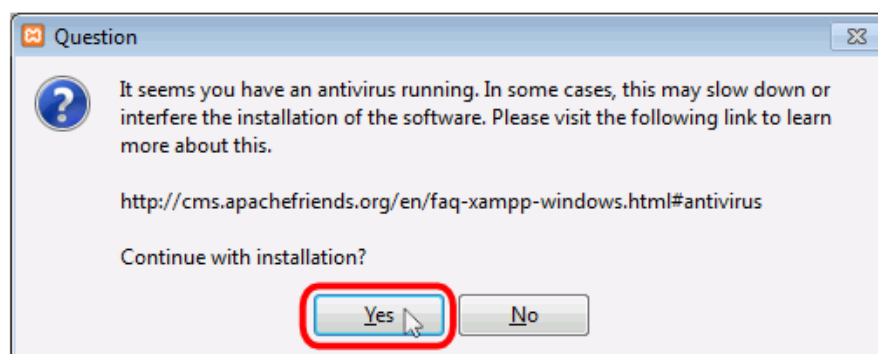


Figura 21 – Aviso de XAMPP por posibles problemas con el antivirus

Una vez iniciada la instalación, hay que con ir pulsando el botón “Next” hasta llegar a la ventana que se muestra a continuación:

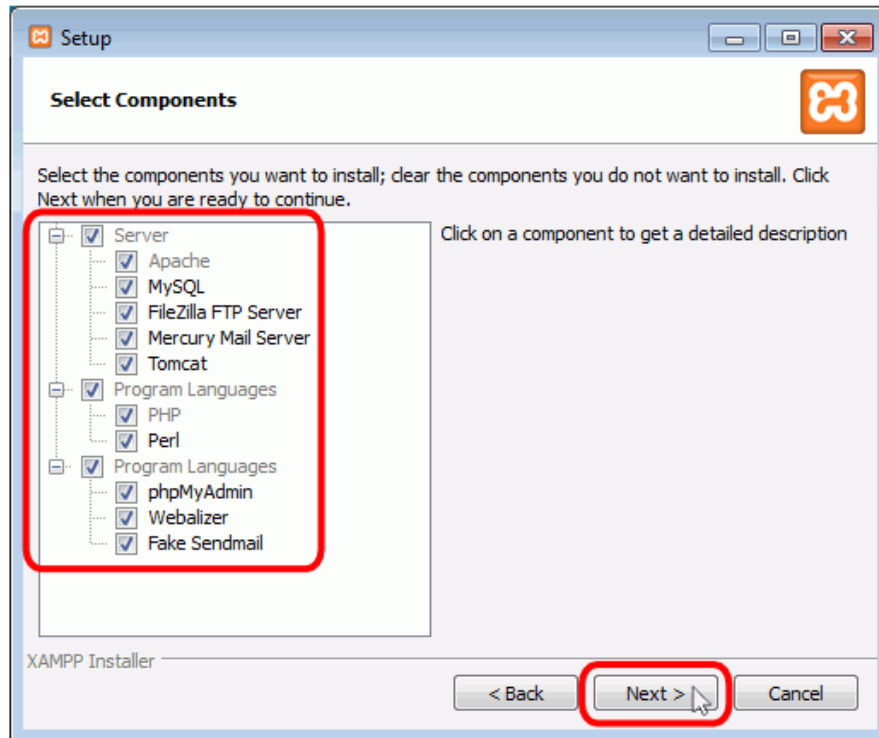


Figura 22 – Ventana selección de componentes XAMPP

Se pueden instalar todos los componentes por si fuesen necesarios en un futuro, pero con instalar Apache, MySQL y PHP es suficiente. Para seguir con la instalación hay que ir pulsando el botón “Next” hasta llegar al final de la misma.

Una vez instalado XAMPP se abrirá el panel de control que se muestra en la Figura 23:

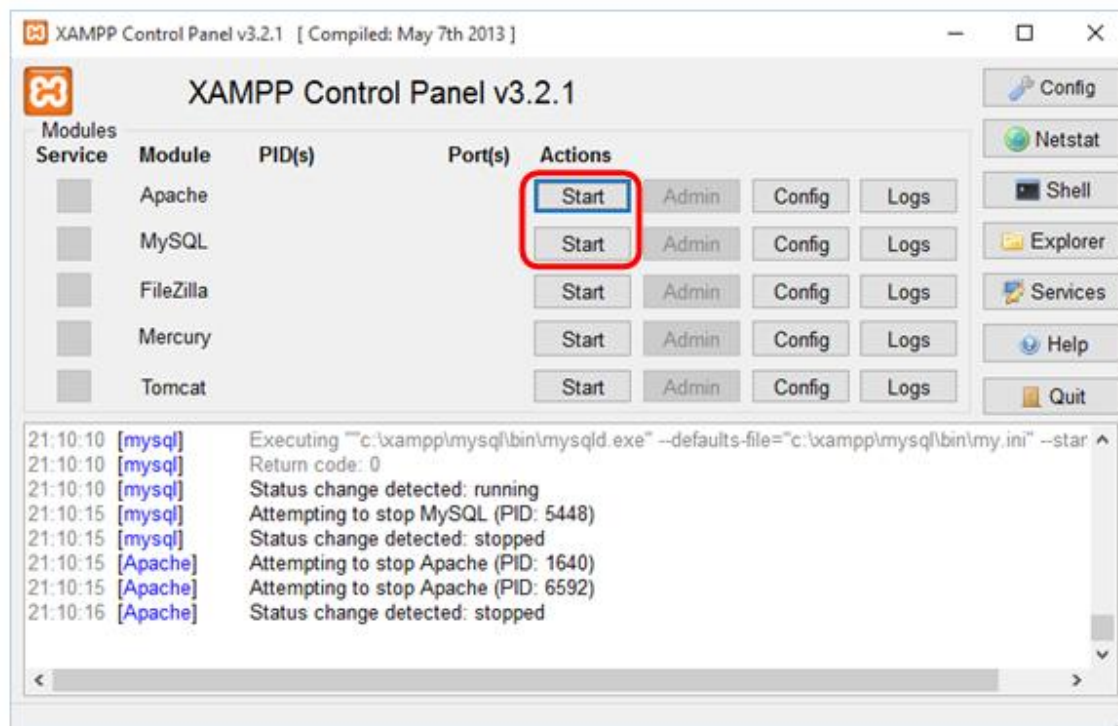


Figura 23 – Panel de control de XAMPP

Una vez abierto el panel de control, es necesario arrancar los componentes Apache y MySQL, para ello hay que pulsar sobre el botón “Start” de cada uno de ellos tal y como puede verse en la imagen anterior.

Si se quiere cambiar la configuración la del servidor, es necesario modificar los ficheros de configuración. Para relizar esta operación hay que pulsar sobre el botón “Config” de Apache y seleccionar el fichero que desee.

En la figura 24 se muestra como acceder a dichos ficheros:

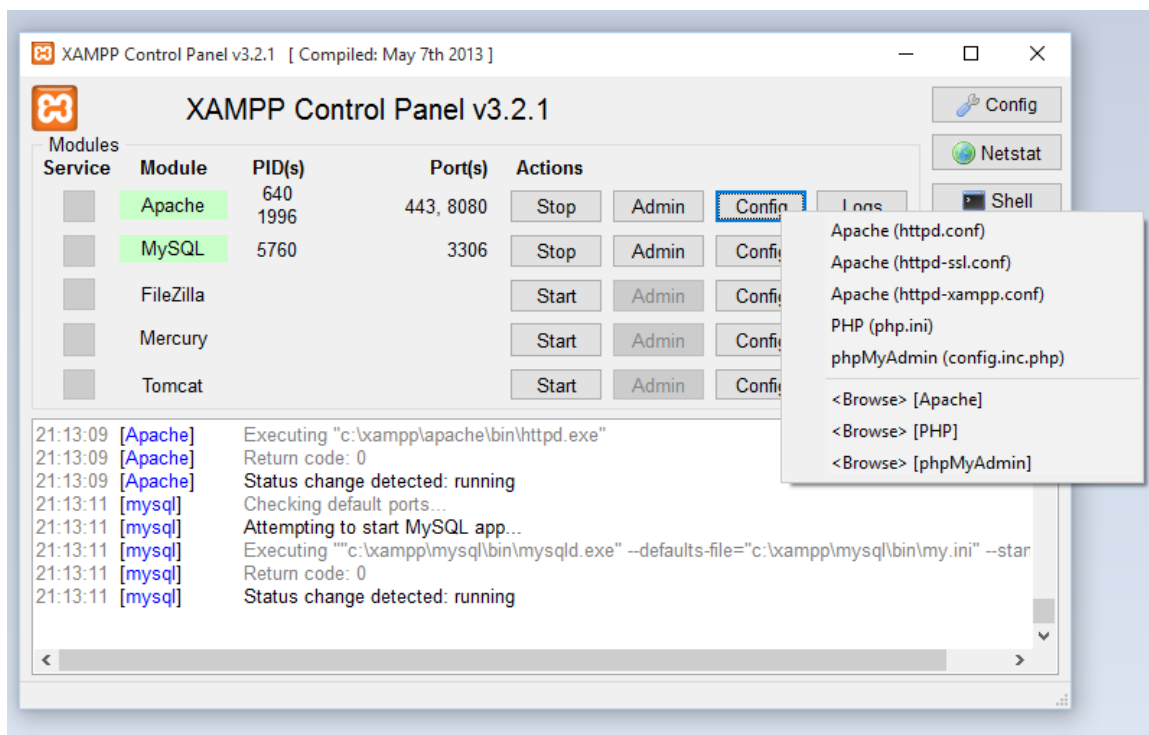


Figura 24 – Archivo Configuración Apache

3.2 Importación de la base de datos

Tras haber instalado y arrancado Apache y MySQL, es necesario importar la base de datos del proyecto. Para ello, se pulsa sobre el botón “Admin” de MySQL y se accede a la consola web de administración de la base de datos tal y como se muestra en la figura 25:

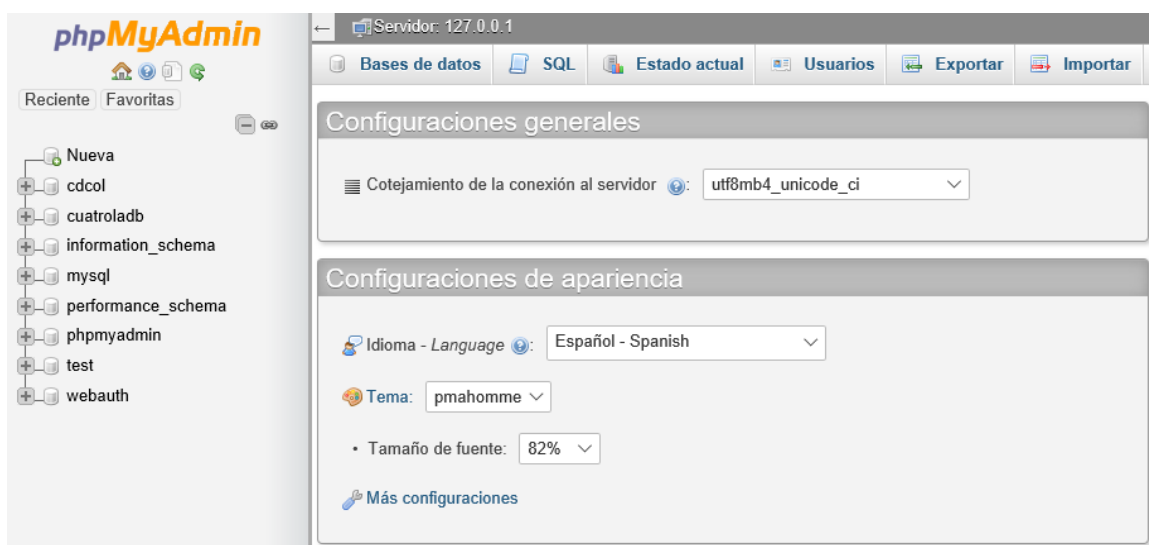


Figura 25 – Consola web de administración de MySQL

Una vez que se ha accedido a la consola de administración, hay que importar el fichero “cuatroladb.sql” con la base de datos. Para importar la base de datos se debe pulsar sobre la pestaña “Importar” y después seleccionar la ruta donde se tenga almacenado el fichero. En la siguiente figura se muestra cómo proceder para la importación de la base de datos:

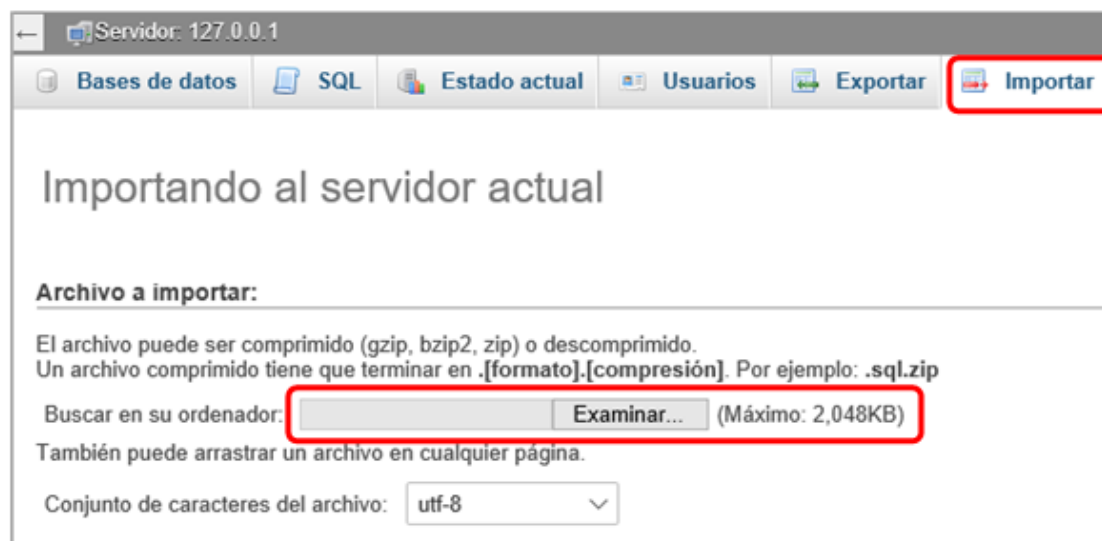


Figura 26 – Importación de fichero en la consola de MySQL

Para finalizar la importación es necesario pulsar sobre el botón “Continuar”. Cuando termina el proceso se muestra una pantalla con el estado del mismo, mostrando los errores ocurridos durante la importación.

3.3 Importación de la base de datos

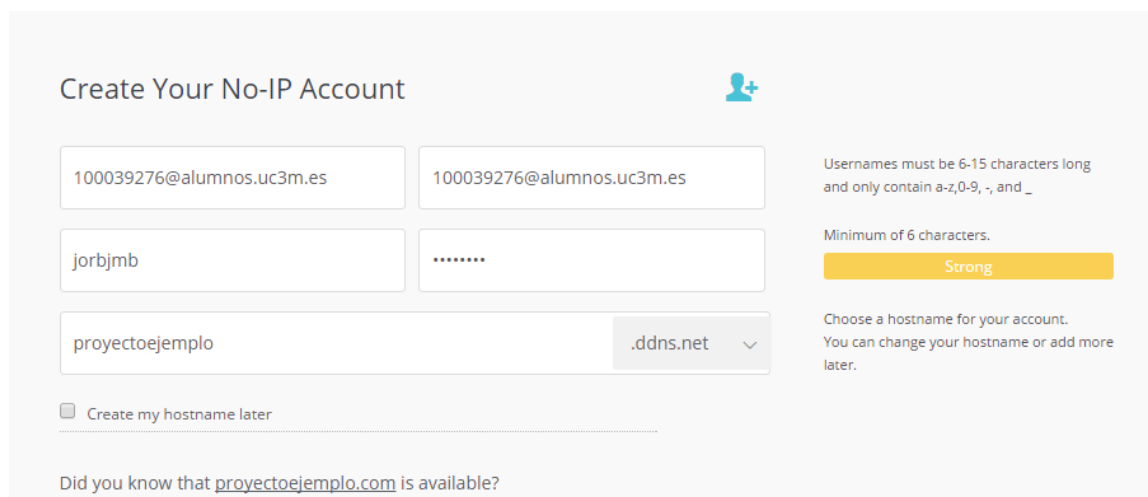
A parte del servidor y la base de datos, es necesario importar los ficheros “.php” con el código a ejecutar.

En la ruta de instalación (por defecto C:\xampp) se crean una serie de carpetas y ficheros con el contenido de la aplicación. En la carpeta htdocs (c:\xampp\htdocs) debe crearse una carpeta llamada “cuatrola” y dentro de la misma hay que almacenar los diferentes ficheros con extensión “.php” adjuntados en el proyecto.

3.4 Configuración e instalación de No-Ip

Para que el servidor sea accesible desde Internet y evitar complicaciones con la configuración del router, es necesario utilizar el proveedor de DNS dinámico No-Ip.

Lo primero que se necesita es crear un host desde la web: <http://www.noip.com/remote-access>. Una vez dentro, hay que rellenar los campos solicitados:



Create Your No-IP Account

100039276@alumnos.uc3m.es 100039276@alumnos.uc3m.es

jorbjmb

proyectoejemplo .ddns.net

☐ Create my hostname later

Did you know that proyectoejemplo.com is available?

Usernames must be 6-15 characters long and only contain a-z, 0-9, -, and _

Minimum of 6 characters.

Strong

Choose a hostname for your account. You can change your hostname or add more later.

Figura 27 – Pantalla registro No-Ip

Hay que tener en cuenta que el nombre rcumplimentado en el campo “hostname”, será el nombre que habrá que utilizar para acceder al servidor desde internet, lo que implica que la aplicación cliente deberá configurarse con este nombre de host.

Una vez cumplimentado el registro, se pulsa sobre el botón “Free Sign Up”. Cuando el proceso se haya finalizado se recibirá un correo de activación en la cuenta de correo indicada en el proceso de registro.

Cuando la cuenta ha sido activada, hay que descargar el cliente No-Ip en la dirección web: <https://www.noip.com/download?page=win> y posteriormente instalar el programa. Durante el proceso de instalación aparece una ventana de configuración como la que se muestra a continuación:

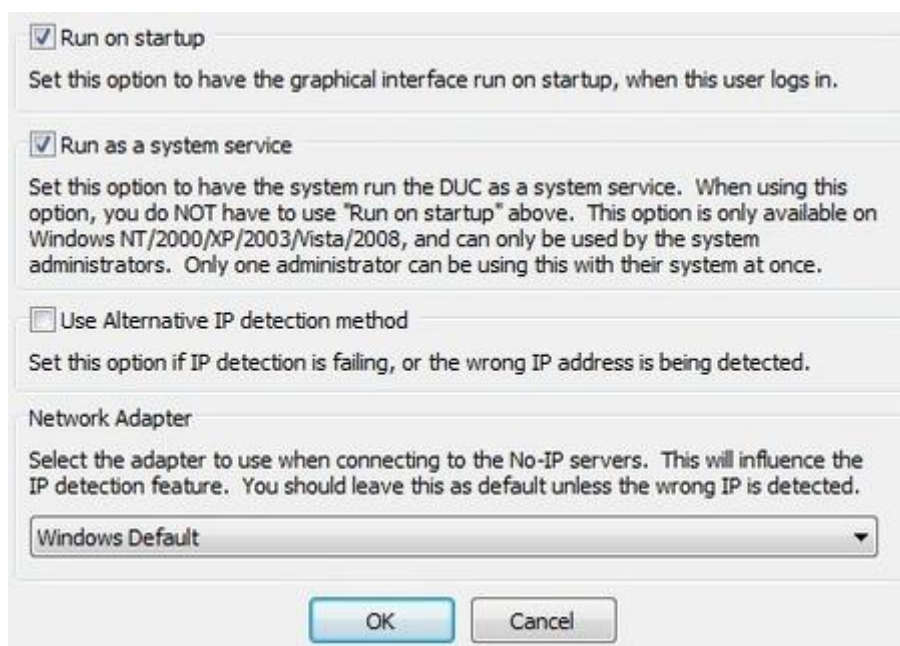


Figura 28 – Ventana configuración No - Ip

Es importante marcar la opción “Run on startup” si se desea que el cliente arranque cuando se encienda el ordenador.

Tras finalizar la instalación es necesario logarse con la cuenta de correo y la contraseña introducida en el registro. Después, hay que seleccionar el nombre de host que se quiera asociar a la dirección ip pública del ordenador, en caso de usar una cuenta gratuita solo se dispondrá del host creado en el registro. En la figura siguiente se muestra un ejemplo de cómo asociar el host:

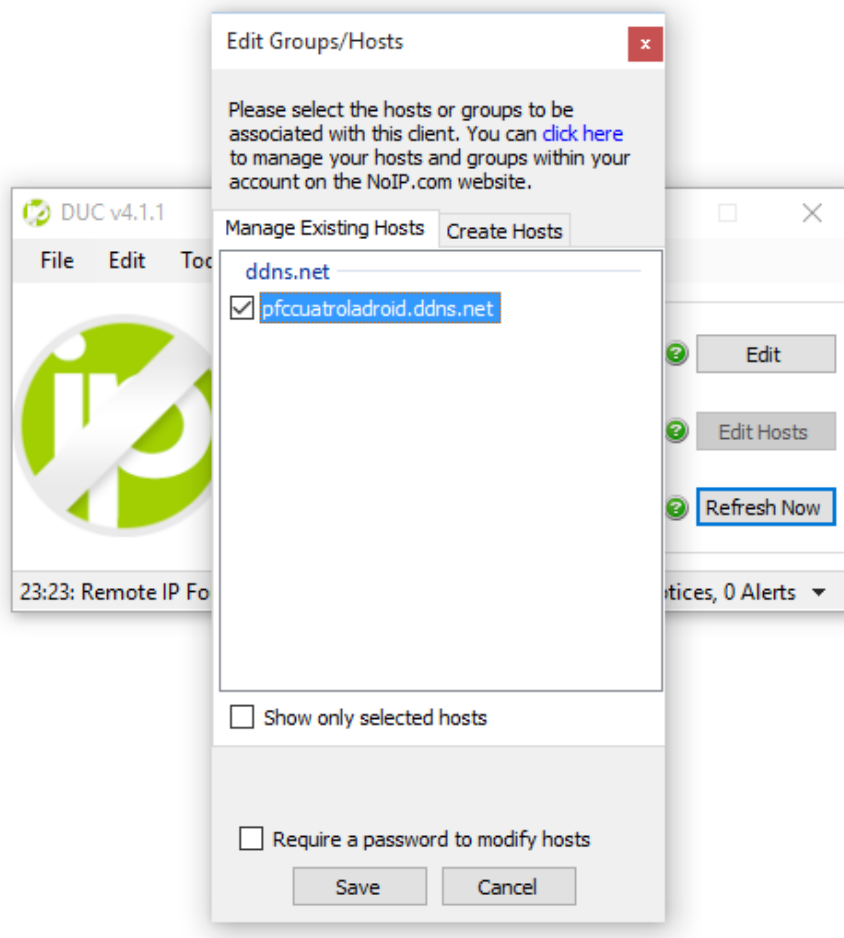


Figura 29 – Asociar host en No-Ip

Cuando se haya terminado con la instalación y configuración de No-Ip, aparecerá la siguiente ventana:

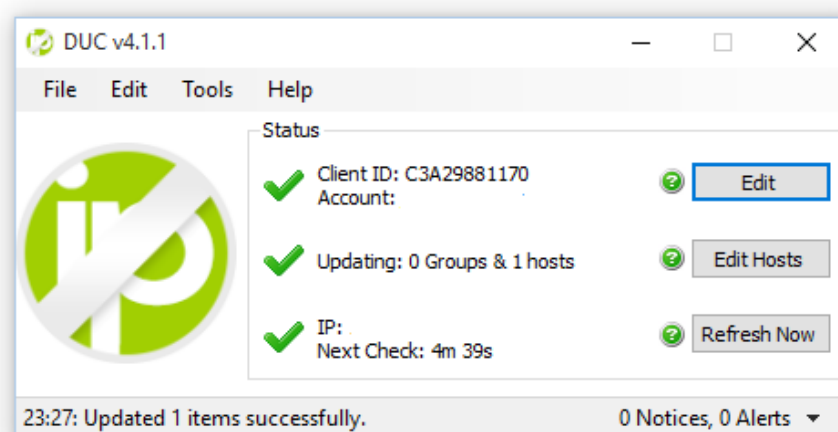


Figura 30 – Cliente No-Ip arrancado y configurado

ANEXO 2: MANUAL DE INSTALACIÓN Y CONFIGURACIÓN DEL SERVIDOR

Tras finalizar este proceso, la aplicación servidor está totalmente operativa para poder proveer los servicios solicitados a las aplicaciones cliente, con lo que los usuarios podrán disfrutar de una partida de Cuatrola.

Anexo 3: Plantillas

1. Plantilla para la definición de casos de uso

CU-XX (identificador de caso de uso)	
Nombre:	Nombre del caso de uso
Descripción:	Descripción del caso de uso
Actores:	Actores que participan en el caso de uso
Precondiciones:	Condiciones que debe cumplir para poder iniciar un caso de uso
Postcondiciones:	Estado en que se queda el sistema tras ejecutar el caso de uso
Flujo normal:	Pasos a seguir para iniciar el caso de uso y pasos que realiza el caso de uso.
Flujo Alternativo:	Caminos alternativos al flujo normal

Tabla 26 - Plantilla para la definición de casos de uso

2. Plantilla para la especificación de requisitos de software

3.

Requisitos Software				
Id	Nombre	Descripción	Necesidad	Prioridad
Identificador requisito	Nombre del requisito	Descripción del requisito	Importancia del Requisito	Preferencia del requisito

Tabla 27 - Plantilla para la especificación de requisitos de software

4. Plantilla para la especificación de pruebas de aceptación

Pruebas de aceptación			
Id	Requisitos probados	Entrada	Salida
Identificador de prueba	Requisitos que se verifican con la prueba	Datos o procesos de entrada	Datos o procesos de salida

Tabla 28 - Plantilla para la especificación de pruebas de aceptación

Resultados pruebas de aceptación		
Id	Requisitos probados	Resultado
Identificador prueba	Requisitos verificados en la prueba	Resultado de la prueba

Tabla 29 - Plantilla para los resultados de las pruebas de aceptación